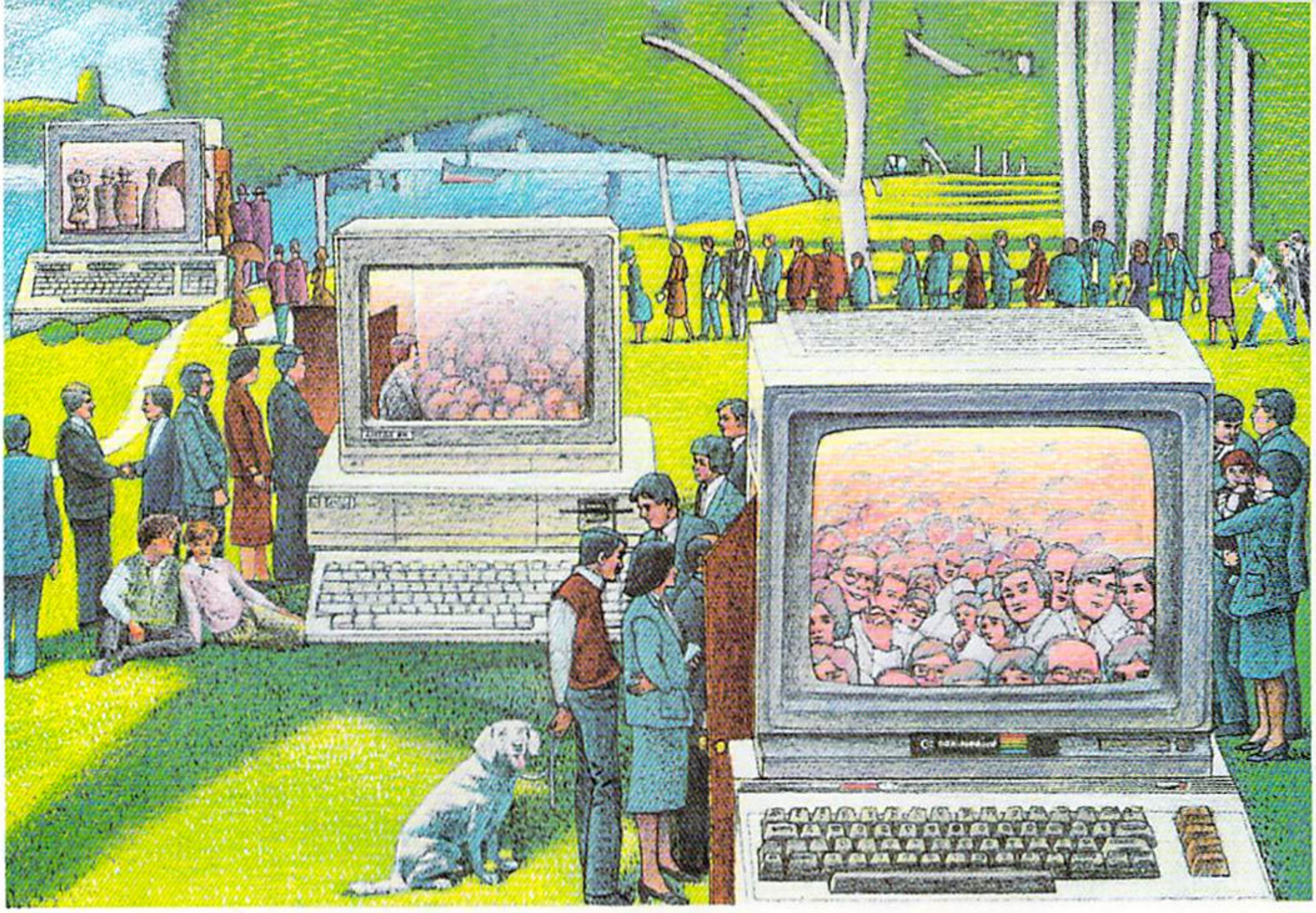
COMMODORE USERS

48355

Ahoy!

AUGUST 1987





USE THE BRAINS YOUR COMMODORE WASN'T BORN WITH.

Right at your fingertips in CompuServe's Commodore® Forums.

Our Commodore Forums involve thousands of Commodore users worldwide. These forums show you just how easy it is to get the most from your Commodore computer.

The Commodore Arts and Games Forum is for all Commodore 8-bit computers, concentrating on music, graphics, and games. The Commodore Communications Forum has updates of communications software. The Commodore Programming Forum supports programmers and developers of Commodore 8-bit computers. And the Amiga® Forum serves as the national resource for all business and entertainment applications in the Amiga community.

Easy access to free software, including free uploads.

You can easily download first-rate, non-commercial software and utility programs. Upload your own programs free of connect time charges. And take advantage of CompuServe's inexpensive weeknight and weekend rates, when forums are most active and standard online charges are just 10¢ a minute. You can go online in most areas with a local phone call. Plus, you'll receive a $25.00 Introductory Usage Credit when you purchase your CompuServe Subscription Kit.

Information you just can't find anywhere else.

Use the Forum Message Board to exchange mail with fellow members. Join ongoing, real-time discussions in a Forum Conference or communicate with industry experts. Scan Forum Data Libraries for non-commercial software. Enjoy other useful services too, like electronic editions of popular computer magazines.

All you need is your Commodore computer (or almost any other personal computer) and a modem.

To buy your Subscription Kit, see your nearest computer dealer. Suggested retail price is $39.95. To receive our free brochure, or to order direct, call 800-848-8199 (in Ohio and Canada, call 614-457-0802). If you're already a CompuServe subscriber, type GO CBMNET (Commodore Users Network) at any ! prompt to see what you've been missing.

CompuServe®

Information Services, P.O. Box 20212
5000 Arlington Centre Blvd., Columbus, Ohio 43220

800-848-8199

In Ohio, call 614-457-0802

An H&R Block Company

Reader Service No. 135

Ahoy!

CONTENTS

DEPARTMENTS

A View from the Bridge...*of the August issue of* Ahoy! 7

Scuttlebutt...*room at the top at Commodore—send your resume now.* 8

Tips Ahoy!...*BASIC pointers and other programming info.* 46

Art Gallery...*alfresco offerings from our artistic readers.* 58

Reviews...*of new hardware, software, and cookware (sort of).* 63

Commodares...*bet you can't agonize over just one.* 71

Program Listings...*18-plus pages packed with games and utilities.* 77

FEATURES

Rupert Report: Green Screen Blues *by Dale Rupert** 32

Entertainment Software Section: Action in Outer Space 41

Amiga Section *(contents on page 53)* 53

COMAL Column: The Death of GOTO *by Richard Herring* 69

*Includes program: SCRNSWAP (for the C-128)

PROGRAMS

Route 64 *by Buck Childress* 13

Steeplechase *for the C-64 by Tony Brantner* 15

Banner Print *for the C-128 by Mike Hoyt* 22

Scanner *for the C-64 by John Krutch* 25

Batter Up! *for the C-64 by James C. Hilty* 31

Archer *for the C-128 by Cleveland M. Blakemore* 60

Bug Repellents *for the C-64 and C-128 by Buck Childress* 78

Flankspeed *for the C-64 by Gordon F. Wheat* 79

Cover art by James Regan and Tom Cushwa; photos by Michael R. Davila and Morton Kevelson

President/Publisher
Michael Schneider

Editor
David Allikas

Art Director
JoAnn Case

Managing Editor
Michael R. Davila

Senior Editors
Tim Little
Tim Moriarty

Production Director
Laura Pallini

Technical Editors
David Barron
Bob Lloret

Consulting Editors
Ben Bova
Morton Kevelson
Dale Rupert

Entertainment Editor
Arnie Katz

Art Production
Christopher Carter
Victoria Green
Stacy Weinberg

Bulletin Board SYSOP
B.W. Behling

Circulation Director
W. Charles Squires

Advertising Director
Lynne Dominick

Director of Promotion
Mark Kammerer

Controller
Dan Tunick

Managing Director
Richard Stevens

Advertising Representative
JE Publishers' Representative
6855 Santa Monica Blvd., Suite 200
Los Angeles, CA 90038
(213) 467-2266
Boston (617) 437-7628
Dallas (214) 660-2253
New York (212) 724-7767
Chicago (312) 445-2489
Denver (303) 595-4331
San Francisco (415) 864-3252

ISSUE NO. 44 AUGUST 1987

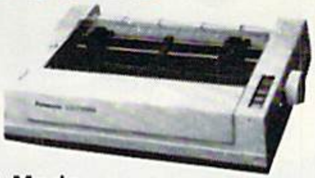
Ahoy! (ISSN #8750-4383) is published monthly by Ion International Inc., 45 W. 34th St., Suite 500, New York, NY, 10001. Subscription rate: 12 issues for $23.00, 24 issues for $44.00 (Canada $30.00 and $55.00 respectively). Second class postage paid at New York, NY 10001 and additional mailing offices.

Postmaster, send address changes to *Ahoy!*, 45 W. 34th Street, Suite 500, New York, NY 10001. Direct all address changes or matters concerning your subscription to *Ahoy!*, P.O. Box #341, Mt. Morris, IL 61054 (phone: 815-734-4151). All editorial inquiries and materials for review should be sent to *Ahoy!*, 45 W. 34th St., Suite 500, New York, NY 10001.

Panasonic

1080*i*

2-year warranty



- 120 cps Draft Mode
- 24 cps NLQ
- Friction Feed

$179.95

Lyco Computer is one of the oldest and most established computer suppliers in America. Because we are dedicated to satisfying every customer, we have earned our reputation as the best in the business. And, our six years of experience in computer marketing is your assurance of knowledgeable service and quality merchandise.

SAVE ON THESE PRINTERS!

INTERFACING

CARDCO

G-WIZ	$39.95
Super G	$49.95

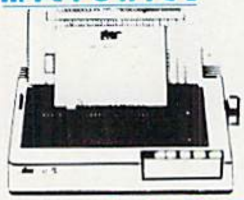
XETEC

Super Graphix (C64) 8K	$64.95
Super Graphix Jr (C64)	$39.95

DSI

PPI	$44.95

star micronics



NP-10 $129.95

- 100 cps Draft Mode
- 25 cps NLQ
- Standard Parallel Interface
- 1-year Warranty

PANASONIC

1080	$179
1091	$249
1092	$319
1592	$389
1595	$449
3131	$359
3151	$399

EPSON

LX86	$CALL
EX-800	$389
EX-1000	$519
LQ-800	$459
LQ-1000	$649
FX-286e	$459
FX-86e	$329

CITIZEN

120-D	$179
MSP-10	$285
MSP-15	$385
MSP-20	$325
MSP-25	$485
Premier 35	$469
Tribute 224	$649

OKIDATA

Okimate 20	$129
Plug n Print	$CALL
120 NLQ	$209
182	$245
192+	$365
193+	$559
292 w/interface	$539
293 w/interface	$679

TOSHIBA

321 P/S	$479
351 sheet feeder	$529
341E	$699
P351 model 2	$1099

STAR M!CRONICS

NX-10	$195
NX-10C	$199
NL-10	$209
SR-10	$399
SD-10	$249
NB-15	$799
ND-15	$425
NX-15	$329
NR-15	$499
NB24-15	$629
NB24-10	$419
ND-10	$329

SILVER REED

EXP 420P	$209
EXP 600P	$539
EXP 800 P	$649

LEGEND

808	$179
1080	$199
1380	$229
1385	$289

DIABLO

D25	$499
635	$779
D-80 IF	$1029

SEIKOSHA

SP-1200 AS	$195
SP-1200 AI	$195
BP-5420	$1075
MP-5300 AI	$549
Color Kit	$119
MP-1300 AI	$369
SL-80 AI	$375
SP-180 VC	$145
SP-180 AI	$145

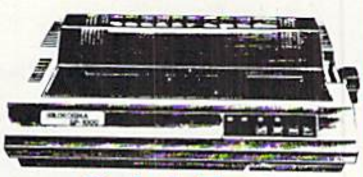
DISKETTES

	5¼	3.5
MAXELL		
SSDD	$9.99	$16.99
DSDD	$12.99	$23.99
BONUS		
SSDD	$6.99	—
DSDD	$7.50	—
SKC		
SSDD	$8.50	$14.99
DSDD	$9.50	$16.99
DSHD	$16.99	—
VERBATIM		
SSDD	$9.99	$16.99
DSDD	$12.99	$24.99

Call for generic diskette pricing!

SEIKOSHA

SP-1000VC



$145.95

Guaranteed Lowest Price!

- 100 cps Draft Mode
- 20 cps NLQ
- Direct Connect C-64/128
- 2-year Warranty

DRIVES

INDUS

GT (C64)	$179

BLUE CHIP

5.25 Drive (C64)	$149
128 Drive	$199

MODEMS

DSI

Messenger	$34.95

AVATEX

1200	$89.95
1200 hc	$99.95 * with cable purchase

MONITORS

TEKNIKA

MJ-305	$309
MJ-503	$529

ZENITH

ZVM 1220 Amber	$89
ZVM 1230 Green	$89

THOMSON

4120 RGB Composite	$249

commodore

C128 PACKAGE $715



- Commodore C-128 Computer
- 1571 Disk Drive
- Thomson 4120 Monitor

Full warranty applies.

commodore HARDWARE

C128 Computer	$249	1802 Monitor	$185
1571 Drive	$229	C1700 RAM	$109
1902 Monitor	$285	C1750 RAM	$169
64C Computer	$175	C1351 Mouse	$39
1541C Drive	$185	1670 Modem	CALL

HOURS

Mon - Thurs 9AM-8PM
Fri 9AM-6PM
Sat 10AM-6PM

Take a Quiet Drive in the Fast Lane.



Introducing the **New**

FSD-2

Excelerator+Plus

Commodore Compatible Disk Drive

Faster, quieter and more reliable than the 1541 and 1541C.
Breakthrough Direct Drive technology.
Guaranteed 100% Compatibility. It even ***enhances*** *GEOS!*
Full One-Year Warranty.

Distributed by: First America Distributing, Inc.
Palmer Lake Plaza Bldg.
6860 Shingle Creek Parkway
Brooklyn Center, MINN. 55430

Manufactured by: **ECI** *Emerald Components International 541 Willamette Eugene, OR 97401 Fax: 503-345-5037 Telex: 23-499-6100 EUG SECSVC*

Reader Service No. 134

VIEW FROM THE BRIDGE

We're probably the only magazine in North America that hasn't featured Vanna White on its cover. While this situation is not likely to be remedied soon, we do have good news for fans of that talented letter-turner—as well as for longtime readers who constantly ask, whatever happened to Bob Lloret?

The author of *Post Time*, *Space Patrol*, and others among the most popular programs of our early issues graces the reverse side of this month's *Ahoy! Disk* with *Fortune Wheel*. This is the most realistic simulation we've seen of the TV show of almost the same name—right down to the dumb blonde! (Er—we meant, Vanna lovers, that she doesn't say much on the show.) If you don't get the *Ahoy! Disk*, see the ad on page 30.

But for our $2.95, the most exciting news this month is right between these covers:

• Tony Brantner's *Steeplechase* and James C. Hilty's *Batter Up!* offer the kind of graphics we consider ourselves fortunate to be able to present to you once per issue. Both are included in this issue. We won't say another word—the screen shots we've printed say everything! (Turn to pages 15 and 31.)

• We know how you hate to type. So before we publish a program that's over five pages in length, we ask ourselves: is this going to be worth it? And in the case of Cleve Blakemore's extended efforts, the answer is usually yes. It certainly is in the case of *Archer*, a multilevel tribute to the arcade milestone called *Venture*. (Turn to page 60.)

• We've run the gamut with banner generators for the 64, from Bob Spirko's mammoth *Streamer Font* (Jan. '86) to Curtis Kaylor's 18-line *Banner* (March '86). Now *Banner Print* by Mike Hoyt lets people who like to print sideways take advantage of the Commodore 128's special features. (Turn to page 22.)

• Because we strive to be to programming what the Mobil Guide is to traveling, *Route 64* seemed like an appropriate title for Buck Childress' latest utility. It facilitates line referencing by replacing calls to line numbers with labels. (Turn to page 13.)

• Encouraged by the fact that no readers called in to report fatal electrocutions sustained from the hardware construction detailed in last month's *Rupert Report*, Dale has gone ahead with an even more ambitious project. This issue's *Green Screen Blues* will allow C-128 users to set up an electronic relay for switching a video monitor from 40 to 80 column mode. (Turn to page 32.)

• While it doesn't incorporate the assembly/disassembly capabilities of full-blown machine language monitors, *Scanner* will allow you to view the contents of any memory location in binary, hex, and ASCII. (Turn to page 25.)

• Because science fiction themes account for such a large segment of the computer game arena, Arnie Katz and Bill Kunkel couldn't cover all the current otherworldly entertainment in a single article. In June they surveyed sci-fi strategy games; this month they provide *Action in Outer Space*, reviewing *Uridium*, *Parallax*, and *Starglider*. (Turn to page 41.)

• If you shun our *COMAL Column* because you have no interest in learning the language, you're making a mistake. Not just because of COMAL's many strong points, but because Richard Herring manages each month to teach as much about programming languages in general as he does about COMAL in particular. This month he makes a case for *The Death of GOTO*. (Turn to page 69.)

Sharp-eyed readers will have noted a slight change in the address we print at various places around the magazine—from Suite 407 to Suite 500. Please use the correct suite number when you write. And when you're in New York, please stop in (as many of you do). We'd love to meet you—and show off our new office! —***David Allikas***

SCUTTLEBUTT

PORTABLE MODEM • MUSIC APPRECIATION DISKS • ENHANCED FORTRAN • VOICE RECOGNITION SOFTWARE • AMIGA CLIP ART • GAMES FROM MICROPROSE, SSI, COSMI, EPYX • CP/M CATALOG • AMIGA CONFERENCES • GUITAR TUTOR • LOTTO PROGRAM • BOOKS

NEW COMMODORE CEO

Commodore has survived so many crises that nothing worries us anymore. Not even the loss of chief executive officer Thomas J. Rattigan, head of the management team that put the company back in the black after the closest brush with bankruptcy in its history. As people who read the financial section know, Rattigan was forced to resign in late April following an unsuccessful power struggle with chairman and chief stockholder Irving Gould. The crux of their disagreement was a difference of opinion on how Commodore US operations should be run, though a clash of personalities also played a role.

The *Wall Street Journal* predicted that the shakeup would continue in the form of job cuts, eliminating up to 40% of Commodore's 200 administrative positions. And this move, the *Journal* speculated, could complicate Commodore's relationship with the banks to which it remains in debt, at a time when the company is reportedly trying to raise financing through a debenture offering due to its banks' refusal to put up more money.

Complicate the relationship it certainly will. But blow it apart? We don't think so. Commodore has come back from too much worse to be derailed by a little executive head-axing, or reaction to same. The company is on a roll —if that fact is not evident to their present banks, it will be to other funding sources. Especially when the Amiga 500 starts selling like Donna Rice buttons at a Republican Convention.

Commodore International, 215-931-4100 (see address list, page 12).

STRING FUNCTIONS

Guitar Wizard ($24.95) helps guitarists learn and analyze the fingerings and fretboard patterns for all types of chords and scales at different positions on the neck. Notes and positions are shown in graphic displays that can be viewed onscreen or printed out. The program can be used with all types of fretted instruments in both standard and non-standard tunings.



For all types of fretted instruments. READER SERVICE NO. 163

Baudville (see address list, page 12).

PICK A WINNER

Of course, you and we know that there's no valid method for predicting winning lottery numbers. But gamblers are a superstitious lot. And for them, *The Lotto Program* ($24.95) will analyze past winning numbers to predict what numbers are likely to come up in the next few drawings. Usable with all 6 and 7 number games, the program produces charts from which you can extract hot and cold numbers, frequency, patterns, sums of digits, digit groups, and more. Because you pick your own numbers from the data generated, your selections won't match those of other owners. For the 64.

Soft-Byte, 513-278-8044 (see address list, page 12).

CP/M SOFTWARE DISKS

Poseidon Electronics' spring/summer catalog of public domain CP/M software is now available. Price of the catalog is $2.00 plus .73 postage; the software costs $16 per disk. Please note, however, that the company will be closed for July and August.

Poseidon requests that any customers who have not received mailings recently contact them, as 50 names were recently lost from their database.

Poseidon Electronics (see address list, page 12).

BOOK RELEASES

Becoming an Amiga Artist ($19.95) tutors beginning and advanced users on the creation of charts, graphs, paintings, tech drawings, digitized images, animation, sound, and speech. Covered topics include programming in Amiga BASIC, reproducing screens on printers, VCRs, and plotters, and working with digitizers, genlock devices, and MIDI interfaces.

Scott, Foresman and Company, 312-729-3000 (see address list, page 12).

The following from Prentice Hall: *The Brady Manual of Spreadsheet Style* ($12.95) gives professional guidelines for creating spreadsheets and developing habits that make spreadsheet analyses faster and more reliable.

The Master Guide to CompuServe ($19.95) saves the user time and money that would otherwise be wasted experimenting online by outlining fast methods of finding desired information. The overall structure of CompuServe is examined, along with features for sophisticated applications and search and retrieval skills.

"SOMETHING TO BYTE INTO"

ABACUS

Basic 128	$38.95
Basic 64	25.75
Cadpak 128	38.95
Cadpak 64	25.95
Chartpak 128	25.95
Chartpak 64	25.95
Cobol 64	25.95
Cobol 128	37.95
Speed Term	25.95
Super C Compiler 64/128	38.95
Super Pascal	48.95

BRODERBUND

Where in the World is Carmen Sandiago	$21.95
P/S Library 1-2-3 ea	15.95
Print Shop	26.95
Print Shop Companion	21.95
Toy Shop	39.95
P/S Holiday Library	15.95

MICROLEAGUE

Box Score/Stat Compiler	$16.95
General Manager Disk	24.95
Micro League Baseball	24.95
Team Disks	16.95

RIBBON

(Min. 6 per order)

Axiom Elite 5 LCD	$ 5.95
Brother 15, 25, 35	5.95
Commodore 1525	6.95
Commodore 1526	6.95
Epson 185	6.95
Epson 85	4.95
Epson mx, fx, rx 100	6.95
Epson mx, fx, rx 80	4.95
Gorilla Banana	6.95
Juki 6000	4.95
Juki 6100	3.95
Legend 880, 1080, 808	6.95
MPS 801	6.95
MPS 803	6.95
OKI 182, 192	7.95
OKI 82, 92, 83, 93	2.25
SG 10-15	2.25
Panasonic 1081, 1091	8.95

INFOCOM

Enchanter	21.95
Hitchhiker's Guide	21.95
Hollywood Hijinx	21.95
Invisiclues ea	6.00
Leather Goddesses of Phobos	21.95
Spellbreaker	26.95
Trinity 128	24.95
Zork 1-2-3 ...... ea	25.95
Sorcerer	24.95

DATAEAST

Commando	$21.95
Karate Champ	21.95
Kung Fu Master	21.95
Tag Team Wrestling	21.95

GAMESTAR

Baseball	$21.95
Basketball 2 on 2	21.95
Football	21.95

PRECISION

Superbase 128	$56.95
Superbase 64	46.95
Superscript 128	47.95
Superscript 64	46.95

TIMEWORKS

Data Manager 128	$42.95
Data Manager II	29.95
Partner 128	42.95
Swift Calc 128 w/Sideways	42.95
Swift Calc 64 w/Sideways	29.95
Word Writer 128 w/Speller	42.95
Word Writer 3 w/Speller	29.95
Partner 641	38.95

DIGITAL SOLUTIONS

Pocket Writer II	41.95
Super Pak 64	34.95
Super Pak 128	44.95

MICROPROSE

Conflict in Vietnam	24.95
Crusade in Europe	24.95
F-15 Strike Eagle	21.95
Gunship	21.95
Silent Service	21.95
Top Gunner	18.95

SPRINGBOARD

Certificate Maker	$29.95
Clip Art I	20.95
Clip Art II	24.95
Clip Art III	24.95
Newsroom	29.95
Graphic Expander	21.95
Certificate Maker#1	21.95

ACCESS

Leader Board	$24.95
L.B. Tournament Disk	14.95
Mach 128	30.95
Mach 5	21.95
10th Frame	24.95
Triple Pack	14.95
World Class Leader Board	25.95

HOT SELLERS

Commando	$21.95
Fist II	19.95
Gato	19.95
Ghosts-n-Goblins	19.95
Kung Fu II	19.95
Parallax	19.95
Power	12.95
Rush-n-Attack & Yie Ar Kung Fu	19.95
Star Fleet 1	29.95
Strike Force Cobra	13.95
Tag Team Wrestling	21.95
Uchi Mata	19.95
Uridium	19.95
X-15 Alpha Mission	19.95

AVALON HILL

Superbowl Sunday	$19.95
S.B. Sunday Stat Compiler	16.95
Team Disk	12.95

DATASOFT

Alternate Reality	$24.95
Black Magic	16.95
Crosscheck	19.95
Goonies	18.95
Mercenary	18.95
Mind Pursuit	18.95
Never Ending Story	18.95
Gunslinger	19.95
221B Baker St	19.95
Theatre Europe	21.95

ACCESSORIES

AICP Disks (12)	$ 7.95
Commodore 1351 mouse	45.95
Commodore 1670 modem	135.95
Disk Notcher	5.95
Disk Storage Case (70)	8.95
Drive Cleaner	7.95
Flexidraw Light Pen	74.95
Messenger Modem	35.95
WICO Boss Joystick	11.50
WICO Bat Handle Joystick	15.50
WICO Three Way Joystick	18.95
Xetek Graphic Interface	54.95
Xetek Graphic Jr	34.95

ELA

America's Cup	$22.95
Artic Fox	22.95
Auto Duel	34.95
Bard's Tale	24.95
Bard's Tale II	27.95
Chessmaster 2000	26.95
Lords of Conquest	24.95
Marble Madness	21.95
Mind Mirror	21.95
Moebius	26.95
Movie Maker	24.95
Murder Party	29.95
Orge	26.95
Quizam	21.95
Scrabble	22.95
Ultima I	28.95
Ultima IV	39.95
Ultimate Wizard	22.95

MINDSCAPE

Bop and Wrestle	$18.95
Fairlight	18.95
High Roller	18.95
Indoor Sports	18.95
Infiltrator	18.95
Lords of Midnight	17.95

BERKELEY SOFTWARE

GEOS	$36.95
GEOS Fontpak 1	18.95
GEOS Desk Pack	22.95
Geodex	24.95
Writer Work Shop	31.95

SOLID STATE SOFTWARE

VizaStar 128	$69.95
VizaWrite 128	59.95

ACCOLADE

Dambusters	$18.95
Fight Night	18.95
Hard Ball	18.95
Law of the West	18.95
PSI 5 Trading Comp.	18.95
Kill Until Dead	18.95
Ace of Aces	18.95
Deceptor	12.95
Comics	18.95

ACTIVISION

Aliens	$21.95
Alter Ego	29.95
Cross Country Road Race	19.95
Gamemaker	24.95
Gamemaker Sci/Fiction	14.95
Gamemaker Sports	14.95
Hacker II	21.95
Labyrinth	21.95
Mindshadow	19.95
Murder on the Mississippi	21.95
Music Studio	21.95
Rocky Horrow Show	18.95
Shanghai	21.95
Tass Times in Tonetown	21.95
Titanic	15.95
Transformers	21.95

EPYX

Fast Load	$23.95
Multiplan 64/128	29.95
Programmers Toll Kit	27.95
Vorpal	21.95
World Karate Championship	18.95
World's Greatest Baseball	23.95
Super Cycle	24.95
World Games	24.95
World's Greatest Football	24.95
Movie Monster	24.95
Championship Wrestling	24.95
Destroyer	24.95
Winter Games	23.95

FIREBIRD

Elite	$19.95
Colossus Chess-IV	21.95
Music Systems	24.95
Advance Music Systems	49.95
The Pawn	28.95
StarGlider	24.95
Tracker	24.95

SSI

Battle of Antietam	$29.95
Carrier Force	36.95
Phantasie II	24.95
Rings of Zelphin	24.95
Wizard's Crown	24.95
Roadwar 2000	24.95

SUBLOGIC

Flight Sim II	$34.95
Football	27.95
Jet	27.95
Pure Stat Baseball	34.95
Scenery Disk 1-6 ea	14.95
Scenery Disk # 7	14.95
Stadium Disk (baseball)	14.95
Up Periscope	18.95

CALL US TOLL FREE

Orders 1-800-634-AICP Only

Customer Service
718-351-1864
10AM to 6PM
E.S.T.

AMERICAN INTERNATIONAL
COMPUTER PRODUCTS
A.I.C.P.
P.O. Box 1758
Staten Island, N.Y. 10314

Reader Service No. 132

ORDER INFO

In stock items shipped within 24 hours for prepaid orders (cert. Check, visa, mc) C.O.D. Orders accpeted, please add $4.00, must be cash or cert. check ONLY. NYS residents add 8¼% sales tax. Advertised prices reflect a 4% discount for cash, credit card add 4%. All orders add $3.50 shipping and handling withing the Continental U.S. Personal checks allow 2 wk. for clearance. Availability and pricing subject to change without notice. Free shipping in Continental U.S. on software orders over $150.00. DEFECTIVE EXCHANGE ONLY FOR SAME PRODUCT. Restocking fee on unopened no-defective items returned. Returns accepted within 15 days of shipping date only. All returns require Return Authorization numbers. WE DO NOT GUARANTEE COMPATIBILITY.

Prentice Hall Press (see address list, page 12).

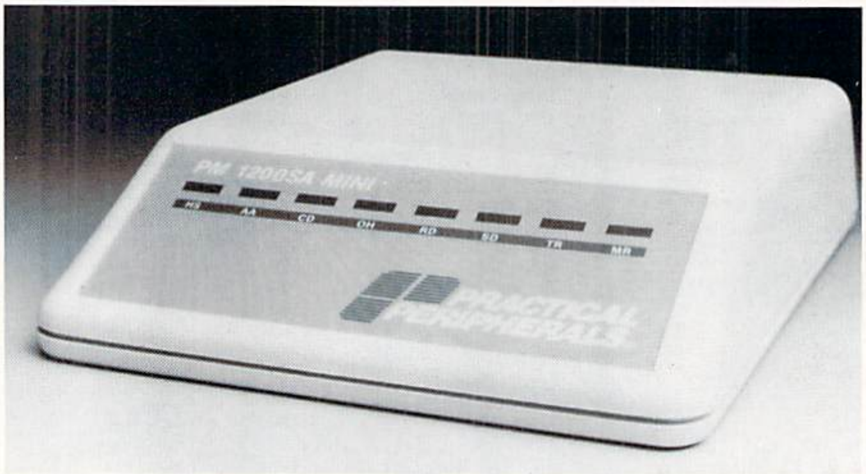
The Bard's Tale II Hint Book ($10.00) contains all 25 dungeon levels plus city maps, and reference sheets for monsters, magical items, and weapons.

Elkon Enterprises, 817-692-2781 (see address list, page 12).

PORTABLE MODEM

The pocket-sized Practical Modem 1200 SA Mini ($199), which retains all the features of Practical Peripherals' full-sized, Hayes-compatible Modem 1200, will plug into your 64 or 128 via an RS232-C interface. The battery-operated portable has a menu-driven configuration (no switches to set) and will store and autodial 10 phone numbers.

Practical Peripherals, 818-991-8200 (see address list, page 12).



Mini modem with 1200 or 300 bps rate and auto dial/auto answer capabilities.
READER SERVICE NO. 162

AMIGA CONVENTIONS

A trio of three-day exhibition/conferences to be held in New York (October), San Francisco (February), and Chicago (May), AmigaEXPO will serve Amiga owners, developers, and dealers with seminars, development forums, and new product displays. For more details, see the ad on page 55 of this issue.

AmigaEXPO Headquarters (see address list, page 12).

WOMEN'S NEWSLETTER

Women Computing divides its 12 monthly pages between female-oriented features (interviews with women executives, female v. male usage statistics) and beginner's articles on topics like computer cleaning, vocabulary, and the like.

They've also got a regular feature called *First Time Tales,* each month profiling a different user's humorous maiden voyage into the field of microcomputing. And that's the main reason we're mentioning the newsletter in *Scuttlebutt.* In the time since *Ahoy!* debuted nearly four years ago, at least two thirds of the people who've bought microcomputers have tried to sell us a humorous account of their maiden voyage into the field of microcomputing. Well, look—there's now somebody soliciting these stories. So don't send them here anymore!

A sample copy is available on request; or, send $36 for 12 issues.

Women Computing, 619-297-7094 (see address list, page 12).

MORETRAN

Pecan has enhanced its *FORTRAN-77* compiler for the Amiga ($99.95) to include support for TYPE COMPLEX, the "ERR =" option, list-directed I/O, support for IEEE standard 2 and 4 word real values, string subset and concatenation routines, and more.

Pecan Software Systems, Inc., 718-851-3100 (see address list, page 12).

CURES LOCKJAW

When teamed with Covox's Voice Master hardware, the memory-resident *Voice Key* voice recognition software allows C-64 users to utilize spoken rather than typed commands with their

programs. Price of the disk and documentation is $29.95.

Covox Inc., 503-342-1271 (see address list, page 12).

FURTHER STRIDERS

More releases in the Strider's Computer Classics line of music appreciation disks, each combining up to an hour of music with as many as 60 screens of commentary on the composers and their work: Viennese Sonatinas (Mozart), Music Antigua (1500's), Bach Favorites, and Sonatinas by Muzio Clementi.

Free Spirit Software, Inc., 312-352-7323 (see address list, page 12).

AMIGA CLIP ART

Two new clip art collections, each $29.95, for use with *DeluxePaint II, DeluxePrint,* and *DeluxeVideo* on the Amiga:

Art Parts: Volume 2 comprises over 125 images and brushes ranging from stars and planets to farm animals to human faces.

Seasons & Holidays has over 100 images covering more than a dozen holidays and special occasions.

Electronic Arts, 415-571-7171 (see address list, page 12).

AMIGA SPELL CHECKING

The *LexCheck* spelling checker ($42.95) works with *Textcraft, Scribble!,* and *Notepad* formatted files, in addition to ASCII text files. The 100,000-word master dictionary resides on disk, enabling the program to use under 100K of RAM so that you can run it simultaneously with your word processor. A two-page document can be checked in under one minute.

Complete Data Automation, Inc., 916-842-3431 (see address list, page 12).

GAMES

Continuing in the neighborhood pick-up game spirit of Epyx's Street Sports line, *Street Sports Baseball* features touches like a field dotted with bushes and tree stumps, and bases made from trash can lids and cardboard boxes. Each player controls pitches, times batters' swings, and regulates field action. For the 64.

Epyx, Inc., 415-366-0606 (see address list, page 12).

Superstar Ice Hockey ($34.95) lets one or two C-64 players take the ice as center or goalie, firing slapshots, hip-checking opponents, and blocking shots, along with coaching their teams and trading and drafting players.

Mindscape, Inc., 312-480-7667 (see address list, page 12).

SSI's first real-time C-64 flight simulator, *B-24* ($34.95) recreates 19 WWII missions flown by the 406th Bombardment Group to the Ploesti, Romania oil fields. The player acts as pilot, co-pilot, navigator, engineer, and bombardier of the lead B-24 Liberator plane, and determines the flight path, speed, altitude, and bomb drop points for the entire formation.

SSI also announced a five-year agreement with TSR, Inc. to produce at least 10 role-playing and several action games based on Advanced Dungeons & Dragons. The games, to be available for the 64 and Amiga among other machines, will begin to appear in early 1988.

Strategic Simulations, Inc., 415-964-1353 (see address list, page 12).

An Amiga (512K) adaptation of MicroProse's *Silent Service* submarine simulation utilizes onscreen icons for all function and activity selections, digitized sound recordings from actual submarine listening devices, and 32-color battle station screens. The program is designed for mouse control, but joystick and keyboard operation is also supported.

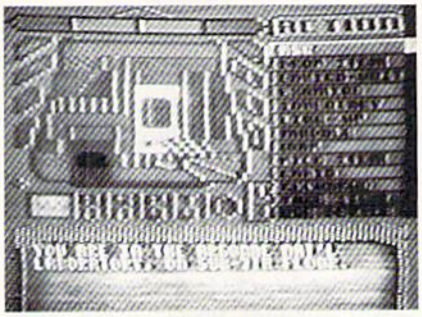
MicroProse Simulation Software, 301-771-1151 (see address list, page 12).

Combining text adventure with scrolling graphics, *Delta Man* ($19.95) takes place in a post-holocaust Chicago. Having survived thanks to your robotic space suit, you pick through the wreckage to find supplies, weapons, and fuel, ward off roaming bands of mutants, and search for other surviving humans until you can escape. For the C-64.

Cosmi, 213-835-9687 (see address list, page 12).

The C-64 gamer must avoid the energy-absorbing stares of *The Sentry* and her Landgazers while attempting through cunning strategy to usurp her as ruler of an alien world. Price is $39.95.

Firebird Licensees (see address list,



Delta Man: ***post-nuclear survivalism.***
READER SERVICE NO. 170

page 12).

Darkhorn ($30.00) allows up to four C-64 strategists to compete simultaneously in real time, utilizing the keyboard, joystick, and paddles in any combination. Each player commands the army of one of the world's four major powers, which meet for a final conflict under the shadow of the towering mountain Darkhorn. Men, elves, and dwarves from each army fan out in all directions, recruiting new members in the villages and forests, while others fortify castles and prepare for the battle. The winner must face the Darklord, who watches from his fortress.

Avalon Hill, 301-254-9200 (see address list, page 12).

As the *Sky Runner* ($29.95) you must save 24th century mankind from the deadly and addictive drug Sky by destroying the huge harvesting machines and their defense towers. For the 64.

Spinnaker Software, 617-494-1200 (see address list, page 12).

Two from Infocom, each $34.95 for the 64, $39.95 for the Amiga:

Infocom's first interactive horror story, *The Lurking Horror* finds you compelled by a mysterious force to rummage through old basements, storage rooms, and decrepit tunnels on your college campus. Included is a mail-in coupon enabling you to purchase *Suspect* for $14.95.

In *Stationfall* you and Floyd (robot star of prequel *Planetfall)* travel on a routine mission to a space station to find it deserted save for an ostrich, an Arcturian balloon creature, and a brainy robot named Plato. The coupon

Continued on page 61

Companies Mentioned in *Scuttlebutt*

AmigaEXPO Headquarters
211 East 43rd Street, Ste. 301
New York, NY 10017

Avalon Hill
Microcomputer Games Div.
4517 Harford Road
Baltimore, MD 21214
Phone: 301-254-9200

Baudville
1001 Medical Park Dr., S.E.
Grand Rapids, MI 49506

Commodore International
1200 Wilson Drive
West Chester, PA 19380
Phone: 215-431-9100

Complete Data Automation
P.O. Box 1052
Yreka, CA 96097
Phone: 916-842-3431

Cosmi
415 North Figueroa Street
Wilmington, CA 90744
Phone: 213-835-9687

Covox, Inc.
675-D Conger Street
Eugene, OR 97402
Phone: 503-342-1271

Delta Research
3867 La Colina Road
El Sobrante, CA 94803
Phone: 415-485-6867

Electronic Arts
1820 Gateway Drive
San Mateo, CA 94404
Phone: 415-571-7171

Elkon Enterprises
2914 Pennsylvania
Wichita Falls, TX 76309
Phone: 817-692-2781

Epyx, Inc.
600 Galveston Drive
P.O. Box 8020
Redwood City, CA 94063
Phone: 415-366-0606

Firebird Licensees
74 North Central Avenue
Ramsey, NJ 07446
Phone: 201-934-7373

Free Spirit Software, Inc.
538 East Edgewood
LaGrange, IL 60525
Phone: 312-352-7323

Infocom, Inc.
125 CambridgePark Drive
Cambridge, MA 02140
Phone: 617-492-6000

MicroProse
120 Lakefront Drive
Hunt Valley, MD 21030
Phone: 301-771-1151

Mindscape, Inc.
3444 Dundee Road
Northbrook, IL 60062
Phone: 312-480-7667

Pecan Software Systems
1410 39th Street
Brooklyn, NY 11218
Phone: 718-851-3100

Poseidon Electronics
103 Waverly Place
New York, NY 10011
Phone: 212-777-9515

Practical Peripherals
31245 La Baya Drive
Westlake Village, CA 91362
Phone: 818-991-8200

Prentice Hall Press
One Gulf + Western Plaza
New York, NY 10023

Scott, Foresman and Co.
1900 East Lake Avenue
Glenview, IL 60025
Phone: 312-729-3000

Soft-Byte
P.O. Box 5701, Forest Park
Dayton, OH 45405
Phone: 513-278-8044

Spinnaker Software
1 Kendall Square
Cambridge, MA 02139
Phone: 617-494-1200

Strategic Simulations, Inc.
1046 North Rengstorff Ave.
Mountain View, CA 94043
Phone: 415-964-1353

Women Computing
2917 Upas Street
San Diego, CA 92104
Phone: 619-297-7094

ROUTE 64

By Buck Childress

Have any of your programs taken a "crash" course lately? The kind that ends with the familiar

```
? UNDEF'D STATEMENT ERROR
```

Or maybe an ill-fated program keeps zipping from point A to point B when it was supposed to connect with point C. A careful check of the list reveals your problem. Now you remember what happened.

While programming this soon-to-be masterpiece, you decided to give your fire-breathing dragon a mop of hair that Godzilla would envy. And, like all good dragons, he enjoys using greasy kid stuff when he combs his tousled head. Naturally you included a gooey sound effect whenever he slops the grease on. Since Ol' Dragon Breath enjoys having the shiniest pate in Lizard Land, that gooey sound will be used a mega-bunch.

You have all kinds of GOTO, GOSUB, and IF-THEN statements just waiting to activate your thrilling 'Goo' noise. Problem is, they all send the program into oblivion. It seems that because you programmed these statements before you programmed the sound effect, you had no idea where to send them. No problem. You assigned each of them a temporary 'dummy' line number. You told yourself that when you finished the Goo routine, you'd go back and change the numbers. Time passed, and so did the memory of those ill-numbered statements. Oh well, all you have to do is locate those dummies and correct them. There's probably not more than 500 of them. What the heck. Lizard Head can hold his fiery breath and slippery comb until you change them. Thank goodness it wasn't a really long program!

Sound familiar? I've definitely had my share of crash courses. If you're tired of your programs taking the wrong turn, put them on *Route 64. Route 64* lets you eliminate references to line numbers and replace them with labels. What the heck are labels?! Just the greatest invention since the mud pie! Whoever originated the idea deserves top billing in the Hacker's Hall of Fame.

Labels save a bundle of time and effort when you're programming. For example, you could have programmed GOTO 'GOO', or GOSUB 'GOO', or IF A=1 THEN 'GOO'. Later, when you told it to, *Route 64* would have changed the labels into the correct line numbers. You can use as many different labels as you want. The more routines you have, the handier *Route 64* becomes.

Remember to save a copy of *Route 64* after you type it in. When run, the loader POKEs the machine language data into memory and checks for errors. If none are found, *Route 64* is ready to use.

Route 64 uses labels that are made up of letters, numbers, or a combination of both, with an apostrophe (') at the beginning and end (i.e., 'GOO' or 'ABC123', etc.). Here are some examples:

```
100 IF A=1 THEN 'GOO'
110 GOSUB 'LOOP'
120 GOTO 'ABC123'
130 A=A+1:ON A GOTO 'GOO', 'LOOP', 'ABC1
23'
```

These are reference labels. As you can see, you just program the way you normally do, using reference labels instead of line numbers. You don't have to put spaces in your program. I did here for clarity. Your labels can be whatever you like. For example, 'LOOP' could refer to a FOR-NEXT loop that you use a lot.

Route 64 only changes reference labels that follow GOTO, GOSUB, and IF-THEN statements that are not part of a REM line, DATA line, or contained in quotes.

When you begin writing a routine that is referenced by a label, just label the routine like this:

```
200 'GOO' REM GOO ROUTINE
```

These are destination labels. A delay loop might look something like this:

```
300 'LOOP' FOR J=1 TO 1000:NEXT J
```

You can also place a destination label on a line by itself:

```
400 'ABC123'
410 REM YOUR ROUTINE BEGINS HERE
```

When you've finished your creation, and want *Route 64* to 'remap' the labels, just type SYS 52000, then press RE-

TURN. Here are the results:

```
100 IF A=1 THEN 200
110 GOSUB 300
120 GOTO 400
130 A=A+1:ON A GOTO 200, 300, 400
```

After *Route 64* remaps your reference labels, it asks if you want the destination labels removed. Pressing the N key leaves them intact and exits *Route 64* via BASIC. You can always have *Route 64* remove them later, if you want. Remember, though, if you run your program and it stumbles across a forgotten label, a syntax error will blow Ol' Laser Breath's fire out.

Press the Y key to remove the destination labels, and

```
200 'GOO' REM GOO ROUTINE
```

will become

```
200 REM GOO ROUTINE
```

By the same token,

```
300 'LOOP' FOR J=1 TO 1000:NEXT J
```

becomes

```
300 FOR J=1 TO 1000:NEXT J
```

If you have a destination label on a line by itself,

```
400 'ABC123'
```

changes to

```
400 :
```

The colon keeps line 400 in your program. All former references to 'ABC123' will still have a route to take.

While it's remapping your masterpiece, *Route 64* checks for errors and, if any are found, tells you what they are. Here are the possible errors:

A) 'Error In Label': forgetting the apostrophes, and/or using characters other than letters and numbers, in your labels.

B) 'Undefined Label': using a reference label without a corresponding destination label.

C) 'Duplicated Label': having more than one destination label of the same name.

D) 'Line Too Long': excessive line length. For example, if the reference label 'A' were being changed to 10000 and, as a result, the line would exceed 80 characters, *Route 64* puts on the brakes.

Route 64 can be brought to a halt at any time by pressing the STOP key.

Let your programming sessions travel on *Route 64*. Mr. Pompadour, your fire breathing comb twirler, will love you for it. □ ***SEE PROGRAM LISTING ON PAGE 87***

STEEPLECHASE

For the C-64





By Tony Brantner

A steeplechase is a horse race which involves jumping over obstacles, such as walls and hedges. In this simulation of the sport, you can race against the clock or another player.

You'll need *Flankspeed* (see page 79) to type in and save a copy of *Steeplechase.* After loading the program, type SYS 49152 and press RETURN to play.

The title screen appears, prompting you to select a course length from one to three miles. Move a joystick plugged into Port 2 to the left or right to change the highlighted selection. When you have made your choice, press the fire button to begin.

The game screen displays two separate windows. The joystick in Port 2 controls the horse in the top window, while the joystick in Port 1 controls the horse in the lower window in a two player race. The elapsed time is shown in the upper left corner of the screen in minutes, seconds, and tenths of a second. At top center is the course length, and in the upper right corner is the lowest elapsed time recorded for the completed course. The distance traveled by each horse is shown, along with a bar meter representing the current speed. The horses enter from the left side of the screen, and when the gun sounds the race begins.

To control a horse, move the corresponding joystick right to increase speed and left to slow down. Smooth scrolling is used in each window to simulate movement. In the background, mountains and clouds scroll slowly, while obstacles in the foreground (stone walls, hills, hedges, and fences) move toward your horse more quickly. Press the fire button to jump over an obstacle. Although some are easy to jump, others require you to build up speed to make it over. Stumbling over an obstacle causes your horse to slow down, so you have to time the jump just right in order to keep your speed high.

The course is randomly created at the beginning of each game. However, in the interest of fair play, both players run the identical course simultaneously. The window of the first player to reach the finish flashes, and if the time is lower than the best so far, it is recorded and displayed during the next game. □*SEE PROGRAM LISTING ON PAGE 88*

PHOTO: MICHAEL R. DAVILA

Reader Service No. 131

COMPUTER DIRECT (A Division of PROTECTO)

SUMMER SIZZLER

Complete C-64 System

INCLUDES: Computer, Drive, Monitor & Printer



SALE

Price Break thru

Sale $399

List $1045

All this for $399

	List
• **Commodore 64 Computer**	$299.00
• **Commodore 1541c Disk Drive**	$229.00
• **Hi-Res 12" Monochrome Monitor**	$249.00
Monitor Cable	$19.95
• **Big Blue Printer w/2 rolls paper**	$199.00
Big Blue Intelligent Interface	$49.95
Total Prices	$1045.90

Complete Atari System

INCLUDES: Computer, Drive, Monitor & Printer



All prices expire 8-31-87

SALE

Sale $399

List $1064

All this for $399

	List
• **Atari 130XE Computer**	$249.00
• **Atari 1050 Disk Drive**	$299.00
• **Hi-Res 12" Monochrome Monitor**	$249.00
Monitor Cable	$19.95
• **Big Blue Printer w/2 rolls paper**	$199.00
Big Blue Intelligent Interface	$49.95
• **Premium Word Processor**	$59.95
Total Prices	$1124.85

160-180 CPS
Hi-Speed Printer Option
With NLQ

Complete Apple® Compatible System

Computer, Drive, Monitor, Printer & Software



Sale

Sale $449*

All this for only $449*:

List $1228

- **Laser 128K Computer**
  video, parallel & serial ports included
  runs virtually all Apple II/IIc/IIe software
- **Hi Res 12" Monochrome Monitor**
- **Big Blue Printer** with interface and paper
- **Software Package:** word processor, spreadsheet & data entry

* Receive $50 Merchandise Discount When System Purchased at $499

Apple® is the registered trademark of Apple Computers Inc.

Complete IBM® XT Compatible System

Computer, Drive, Monitor, Printer & Software



Sale

Sale $599

All this for only $599 :

List $2495

- **10 MHz / 512K Super Turbo XT Computer**
  video, parallel & serial cards included
  3.2 MS DOS & G.W. Basic included
- **Hi-Res 12" Monochrome Monitor**
- **Big Blue Printer** with interface and paper
- **Software Package:** word processor, spreadsheet & data base

IBM® is the registered trademark of International Business Machines, Inc.

Shipping, Handling & Insurance Charges
Add $35.00 for shipping, handling, and insurance. Illinois residents please add 6½% sales tax. Add $70.00 for ALASKA, CANADA, HAWAII, PUERTO RICO & APO-FPO orders. All orders must be in U.S. Dollars. WE DO NOT EXPORT TO OTHER COUNTRIES EXCEPT CANADA & PUERTO RICO. Enclose cashier check, money order or personal check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail. Prices & Availability subject to change without notice.
VISA — MASTER CARD — C.O.D. Call For C.O.D. Charges.

COMPUTER DIRECT
A Division of **PROTECTO**
22292 N. Pepper Road,
Barrington, Ill. 60010
We Love Our Customers

312/382-5244
For Commodore • Atari
312/382-5050
For Apple • IBM

Reader Service No. 131

Reader Service No. 131

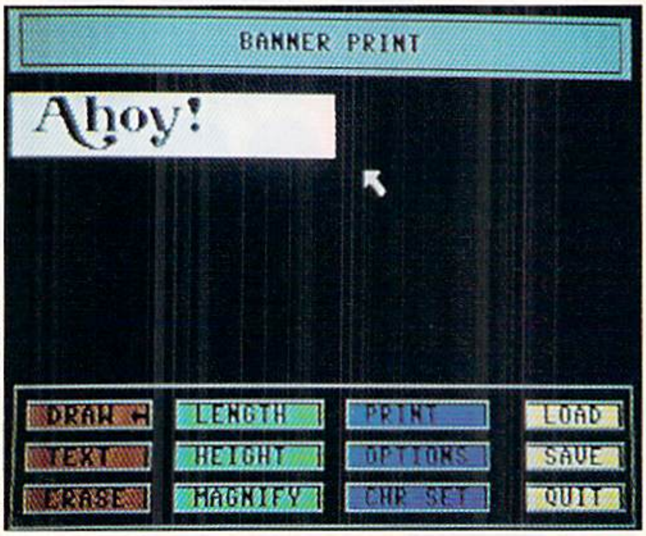
Banner Print

A banner printer for the C128

By Mike Hoyt

B*anner Print* is a joystick-driven utility written for the Commodore 128 in 128 mode. It uses hi-res graphics to let you see your entire banner at any time. You can magnify and edit portions of your banner, add text, draw lines and dots, load or save it, and print it using a variety of options. *Banner Print* was designed to be used with the Okimate 10 or a Commodore 1525-compatible printer; it will work with other printers that can be used with the Commodore, but you may have to sacrifice some of the program's special printing options.

Enter the program and save it to cassette or diskette. After you run the program, you will see the banner, a white strip, across the upper portion of the screen. At the bottom is a box filled with various labeled buttons. In the center of the box is an arrow. This is your pointer; you can control its movement by using a joystick in Port 2. Notice that when the pointer is at the bottom of the screen, it moves faster than when it is near the banner. This allows you to have finer control of the pointer when drawing on the banner. Using either set of arrow keys will also move the pointer, but it will not slow down when it is in the banner editing area. The buttons at the bottom of the screen can be activated by positioning the pointer inside the box and pressing the joystick trigger. You can also draw on the banner by moving the pointer where you want and pressing the trigger. You can draw lines by holding the trigger down while moving the pointer over the banner. When the pointer moves over a pixel that is white, it will turn it black, and vice versa. The black pixels represent areas of your banner that are printed on the printer.



PHOTOS: MICHAEL R. DAVILA

A banner file loaded onto **Banner Print's** *main menu screen.*

Following is an explanation of each of the 12 buttons. Next to the name of each button (in parentheses) is listed an equivalent keypress that can be used to activate the button instead of positioning the pointer and pressing the trigger.

DRAW (D)

Draw does just what it says. Move the pointer to the area of the banner you want to draw on, and press the trigger to draw. If the cursor is over a lighted pixel, it will turn it off. Likewise, if the cursor is over a dark pixel, it will turn it on. To draw long lines, just hold the trigger down and keep moving the pointer in the direction you desire.

TEXT (T)

This function allows you to type text directly onto your banner using the currently selected character set (see CHR SET below). When this button is selected, an arrow will point to the word "TEXT" on the button. Move the pointer anywhere on your banner and press the joystick trigger to begin entering text. A blue box appears where the next character you type will be placed. Use the INST/DEL or the left/right cursor keys to move the box, or press the RETURN key to exit text mode and return control back to the pointer.

ERASE (E)

Be careful about selecting this button! It will erase your entire banner and let you start all over. Fortunately, you will be asked if you are sure of your decision.

LENGTH (L)

This button allows you to select a new length for your banner. Press RETURN to cancel, or enter a number from 1 to 120 to select the length. If the length of the banner is wider than the screen, the banner will "wrap around" to the other side. When working on your banner, remember that if it "wraps around" the screen, it is continued on the left side of the banner strip below it.

HEIGHT (H)

Pressing this button lets you select your banner's height. Enter the new height, from 8 to 24 pixels, or press RETURN to leave the height as it is. In relation to the height

#2—FEB. '84 Illustrated tour of the 1541! Artificial intelligence! Synapse's Ihor Wolosenko interviewed! String functions! And ready to enter: Music Maker Part II! Screen Manipulation! Night Attack! Relative Files!	**#3—MAR. '84** Anatomy of the 64! Printer interfacing for VIC & 64! Educational software series begins! VIC game buyer's guide! And ready to enter: Address Book! Space Lanes! Random Files on the 64! Dynamic Power!	**#5—MAY '84** Future of Commodore! Inside BASIC storage! Memory management on the VIC & 64! Guide to spreadsheets! Hurray for arrays! And ready to enter: Math Master! Air Assault! Biorhythms! VIC Calculator!	**#6—JUNE '84** Game programming column begins! Code generating programs! Rupert on inputting! Memory management continues! And ready to enter: Post Time for the 64 & VIC! Alpiner! Sound Concept!
#7—JULY '84 MSD dual disk drive! Database buyer's guide! Training your cursor! Screen displays! File Sleuth! Users Groups! And ready to enter: Renumbering! Checklist! Math Defender! Brisk! What's My Job?	**#8—AUG. '84** Choosing a word processor! Computational wizardry! Creating your own word games! Sound on the 64! And ready to enter: Micro-Minder! Directory Assistance! The Terrible Twins! Words Worth!	**#9—SEPT. '84** Program your own text adventure! Build a C-64 cassette interface! Video RAM! Word processors, part II! And ready to enter: Salvage Diver! DOS! Sound Explorer! The Castle of Darkness! Base Conversions!	**#10—OCT. '84** C-64 graphics programs! Bit-mapped graphics! Joystick programming! Graphics processing! And ready to enter: VIC 40 Column Operating System! BAM Read & Print! Emerald Elephant! Lawn Job!
#11—NOV. '84 Music programs & keyboards for the 64! Graphics feature continues! 2-D arrays! And ready to enter: PTE word processor! Block Editor! Alternate Character Set for the 64! Tunnel of Tomachon!	**#12—DEC. '84** Buyer's guide to printers! 1525 printer tutorial! Fast graphics with custom characters! User Guide to KMMM Pascal! Diving into BASIC! And ready to enter: Construction Co.! Space Patrol! Cross Ref!	**#13—JAN. '85** VIC and 64 OS exposed! Sprites! Insert a 1541 device # disconnect switch! Ghostbusters! And ready to enter: Ultra Mail! Music Tutor! Alice in Adventureland! Midprint! To the Top! Tape/Disk Transfer!	**#14—FEB. '85** Printer interfacing! Multicolor sprites! Modems! Bulletin Boards! Theory of game design! Buying a modem! And ready to enter: Futurewar! Fontasia! VIC Eraser! Insurance Agent! Flankspeed! Telelink 64!
#15—MAR. '85 Creating multiscreen gameboards! Inside the Plus/4! Commodore DOS! And ready to enter: Old Routine! Programmable Functions! Automatic Line Nos.! Home Budget! Salmon Run! Numerology!	**#16—APR. '85** Assembly language column begins! Programming the joystick! 1541 disk drive alternatives! The Kernal! And ready to enter: Hop Around! Faster 64! Booter! Elecheck! BASIC Trace! Space Hunt!	**#17—MAY '85** Disk drive enhancements! Install a reset switch! Assembler escapades! And ready to enter: Super Duper! Two-Column Directory! DSKDU! Raid! DOS Plus! Font Editor! Tile Time! Interrupt Wedge!	**#18—JUNE '85** Music & graphics entry system! How modems work! Inside the 6510! And ready to enter: Quad-Print! Mapping 4.4! Towers of Hanoi! Speedy! Duck Shoot! Bit Dumping! Screen Magic! 6510 Simulator!
#19—JULY '85 PROM Programming! 3-part harmonies on VIC/64! Speeding pixels! And ready to enter: Auto-Append! Script Analysis! Wizard of Im! Lucky Lottery! Brainframe! Etch! Printat! Autos: Leasing v. Buying!	**#20—AUG. '85** Inside the 128! Real-world simulations! Sound effects! Modems! And ready to enter: Windows! Formatter! Sound-a-Rama! Screen Dump! Selectachrome! Disintegrator! Fidgits! Gators N Snakes!	**#22—OCT. '85** Create cartoon characters! Infinitesimal intrigue! Secrets of copy protection! And ready to enter: Shotgun! Maestro! Solitaire! Mystery at Mycroft Mews! Gravinauts! 1541 Cleaning Utility! Shadey Dump!	**#23—NOV. '85** Guide to adventure gaming! ML sprite manipulation! BASIC for beginners! And ready to enter: Lightning Loader! Knight's Tour! Chopper Flight! Rhythmic Bits! Instant Bug Repellent! File Scout! Slither!
#24—DEC. '85 Speech synthesizer! The IBM Connection! The year's 25 best entertainments! And ready to enter: Gypsy Starship! Directory Manipulator! Cloak! Gameloader! Jewel Quest! Lineout! Santa's Busy Day!	**#25—JAN. '86** Build a speech synthesizer! Survey of sports games! And ready to enter: Martian Monsters! Streamer Font! Microsim! Haunted Castle! Knockout! Infraraid! Alarm Clock! Memory Check! Scratch Pad!	**#26—FEB. '86** Windows! Build an auto-exec cartridge! Align your 1541! Survey of flight simulators! Structured programming! And ready to enter: Arena! Head to Head! Crabfight! Treasure Wheel! Character Dump!	**#27—MAR. '86** Programming educational games! Memory dumpers! Choosing a copy program! Custom characters! And ready to enter: Ahoy!Term 128! Trivia Game Maker! Brickbusters! Easy Lister! Programmer's Aid!
#28—APR. '86 Comet catching! Survey of action and strategy games! Screen dumping! And ready to enter: Chrono-Wedge! Mr. Mysto! Air Rescue! Notemaker! Screen Window! JCALC! Hidden Cavern! Swoop!	**#29—MAY '86** 128 graphic bit map! Epyx strategy guide! 128 commands! ML music programming! And ready to enter: Bigprint! Star Search! Failsafe! English Darts! Ski Folly! Free RAM Check! Alchemist's Apprentice!	**#30—JUNE '86** Debugging dilemmas! Public domain software! Winning at Ultima! Computer Aided Design! And ready to enter: LazyBASIC! Got A Match? Star Strike! Queen's and Bishop's Tours! Shaker! Trackdown!	**#31—JULY '86** Inside the Amiga! Conditional branching! Chess programs! 128 and 64 DOS! And ready to enter: Screen Sleuth! Skull Castle! Head-on! Nebergall Run! Wordcount! Crazy Joe! Fidgits' Music School!
#32—AUG. '86 Inside the Amiga, part II! Approaching infinity! C-64 war simulations! Pascal for beginners! ML graphics! And ready to enter: Reversi! Highlight! Disk Cataloger! Meteor Run! Trim! Step On It! Flap!	**#33—SEPT. '86** Windows and viewports! Sound & music on the 64! COMAL! And ready to enter: The Last Ninja! Speech64! Multi RAM! Dogcatcher! Trapped! Matchblocks! Variable Manager! Dual Dump! Mine Canyon!	**#34—OCT. '86** Build a digital oscilloscope! ML speed techniques! And ready to enter: Vault of Terror! Quick Change! Penguins! Attack Force! Disk Checkup! Dvorak Keyboard! Mountaineer Mack! 128 to 64 Autoboot!	**#35—NOV. '86** C-128 shadow registers! Data file handling! PROMAL! Habitat! And ready to enter: Teleporter! 128 RAM Check! Discs of Daedalus! Guardian! Tenpins! Syntax Patrol! Deluxe List! Long Lines! Detonation!
#36—DEC. '86 File manipulation! C-128 shadow registers! Football games! And ready to enter: The Artist! Minotaur Maze! Mouse in the House! Lazy Source Code! Rebels and Lords! Speedway! The Editor! Micro City!	**#37—JAN. '87** Pointers and the monitor! Best games of '86! DOS for beginners! And ready to enter: Vortex! Hanger 14! BASIC Ahoy! Catacombs! Lixter! Dark Fortress! PermaLine! Starfighter! Bugout! Screens!	**#38—FEB '87** Hacking into machine language utilities! Amiga RAM expanders! And ready to enter: Window Magic! Crunchman! User Conventions! The Adventurer! More BASIC 128! Jailbreak! Turtle Rescue! 640!	**#39—MAR. '87** Basic esthetics! Survey of video digitizers! Multiplayer games! And ready to enter: C-64 Compressor! Wizard Tag! Turbopoke! Rescue 128! Lights Out! Pinball Arcade! Stow Away! Caverns of Geehonk!
#40—APR. '87 Inside the Amiga 2000! Fractals! Baseball games! COMAL, turtle graphics, and Logo! And ready to enter: Infoflow! Laps! Pieman! List Formatter! Scrambler! Extended Background Mode! Planet Duel!	**#41—MAY '87** Kernal power! 64 and Amiga graphics! Microworlds in COMAL! Brain games! Dark Fortress master maps! And ready to enter: Moondog! Startup! Illusion Master! Wall Crawler! Scavenger Hunt!	**#42—JUNE '87** Megaflops and microseconds! Sci-fi braingames! C-64 to Amiga file transfer! And ready to enter: D-Snap! Wraiths! Galactic Cab Co.! Cave of the Ice Ape! ALT-Key 128! Power Squares! 128 Multi RAM!	**#43—JULY '87** Real world interfacing! Bit map graphics tutorial! C-64 graphic conversion! Martial arts software! And ready to enter: Wizard Tag II! Data Creator! Plink & Plonk! Univaders! Data Express! 128 Scroller!

BACK ISSUES OF Ahoy!

Why settle for reading *Ahoy!* only once a month? Order the back issues listed above and fill up those boring in-between weeks with all the programs, articles, and reviews you've missed! Use the coupon below to order your issues while limited supplies last.

Use coupon of facsimile. If ordering more than four issues, list choices on a separate sheet.

If you have a modem and want more complete information on any of the back issues listed, call *Ahoy!'s* Bulletin Board Service at 718-383-8909.

Please send me

____ copies of issue #____ ____ copies of issue #____

____ copies of issue #____ ____ copies of issue #____

ALL ISSUES $4.00
EXCEPT #2 AND #3 ($10.00)

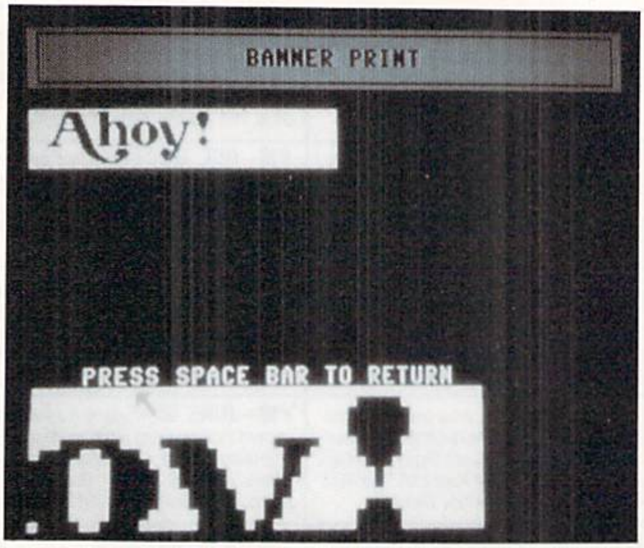
Enclosed please find my check or money order for $______ (outside the USA add $1.00 per copy)

NAME________________________________

ADDRESS________________________________

CITY________________ STATE________ ZIP________

Send to: *Ahoy!* Back Issues, Ion International Inc., 45 West 34th Street—Suite 500, New York, NY 10001.



Magnify mode enables you to draw and erase pixels on a magnified portion of your banner, just as in draw mode.

you enter, the banner will grow in height on the screen.

MAGNIFY (M)

This control allows you to do detailed work on a portion of your banner. A green rectangle will appear at the upper left corner of your banner. Using the joystick, position the rectangle over the area you want to magnify. After pressing the trigger, the screen will blank while the 128 magnifies the selected area. After a few seconds, the banner display will return, and the pointer will be positioned in the lower left area of the screen over the magnified view. You can draw and erase pixels on the magnified view as you do when using the draw mode. Once you have finished, press the space bar and the control panel will reappear.

PRINT (P)

When you select this button, the program will ask you to prepare your printer and press the space bar to begin printing. While the banner is being printed, pressing the space bar stops the printer and gives you back control over the pointer. If you press "F" while the banner is printing, you can enter FAST mode. In FAST mode the screen blanks and the computer speeds up to 2 Mhz, thereby taking much less time to print your banner. Press "F" again to return to the normal "slow" speed.

OPTIONS (O)

Pressing this button allows you to choose options for printing your banner. The seven options are:

1. Print Graphic Blocks: If you answer "Y" to the prompt, the banner will be drawn using graphic blocks (options 2 and 3 will be skipped). Note that this option may not work on non-1525 compatible printers.

2. Print Character: Press any character from the keyboard to use for printing the banner. If your printer supports the Commodore graphic character set, you may use any graphic character on the keyboard.

3. Fine Line Spacing: Answer "Y" to this prompt and your printer will squeeze lines of text together. "N" will let the printer do normal linefeeds. This option may not work on non-1525 compatible printers; if in doubt, consult your printer manual.

4. Expand to Width of Page: Reply with "Y" if you want to s-t-r-e-t-c-h the vertical pixels to take up an entire 80 column print line. Selecting this skips option 5, the pixel height option, since you have allowed the program to select the height for you.

5. Pixel Height: This will let you select the number of columns wide each pixel from the banner will be on paper. Note that the banner is rotated 90 degrees when printed, so pixel height refers to the height on the screen and the width on paper.

6. Pixel Width: Enter a number from one to ten to select how many rows make up each pixel on paper. The higher the number, the longer your banner will be.

7. Special Printer Codes: If your printer allows special options that you want to use, such as a bold or italic character font, enter the ASCII equivalent of the code stated in your printer's manual. If the code is made up of more than one ASCII code (an escape sequence, for example), you can continue entering codes until you type a "0", which signifies the end of the codes. For example, my Okimate 10 defaults to skipping the perforation on continuous form paper. This feature would cause gaps in parts of my banner, so I would enter 27 (RETURN) 66 (RETURN) 0 (RETURN) at the prompt to disable the perforation skip, which is the ASCII equivalent to ESC B, the sequence stated in my manual.

CHR SET (C)

This control allows you to select the uppercase/graphics set or the lower- and uppercase character set.

SAVE (S)

This will save your banner as a binary file. You can type up to 12 characters as the filename or just press RETURN to cancel the save. When you enter the filename, ".BNR" (BaNneR) is appended to it to distinguish the file from other programs or data files. Note that both the LOAD and SAVE features require a disk drive. Cassette users will either have to modify the load and save routines or do without these features.

LOAD (G)

Select this to load a banner file. RETURN cancels the load. The ".BNR" suffix is automatically appended to the end of the filename.

QUIT (Q)

Select this button when you've had enough! A prompt will ask you to confirm your decision in case you accidentally selected QUIT.

The size of the final version of *Banner Print* surprised me. It would have been at least twice as long without the powerful graphics and sprite commands provided in BASIC 7.0. I think *Banner Print* should prove to be a useful utility for your Commodore 128. ☐

SEE PROGRAM LISTING ON PAGE 84

SCANNER

A Beginner's Machine Language Monitor for the C-64

By John Krutch

Scanner, a Commodore 64 machine language monitor, was designed for beginners, though more advanced users may find it helpful as well. It's the kind of monitor that I wish I'd had when I was beginning with the 64.

Scanner lets you look at any location in memory and observe its contents in binary and hexadecimal and as an ASCII character. *Scanner* is graphics-oriented. Memory locations are represented as cells, and *Scanner* itself is represented as a window above the memory cells. You can scan forward or backward through memory just by pressing a key.

A handy feature of *Scanner* is that its display is continuously updated. Each memory cell on the screen is updated 15 times per second, making it easy to see what's happening in memory locations that change as you press various keys or which change for some other reason. This is especially useful for examining the lowest 1K of memory, where there are many locations whose contents change frequently to reflect changes in the system.

Flankspeed (page 79) is required to enter *Scanner.* When you've finished making a copy, use LOAD "FILENAME",8,1 to load from disk or LOAD "FILENAME",1,1 to load from tape. Then use SYS 49152 to start the program.

To use it, just remember that *Scanner* is a moving window that lets you scan up and down through memory. Press f1 to move the window up one memory cell (toward high memory). Keep f1 pressed to scan up continuously. Press f3 to move the window down one memory cell (toward low memory). Keep f3 pressed to scan down continuously.

To switch the window to a new group of memory cells, press f7. A panel with a cursor will appear at the top of the screen. Type an address using four hex digits. For example, to see zero page location $E5 type

00E5

The memory cell plus the next four cells will appear.

The address of the memory cell is the four-digit hexadecimal number on the dark blue field; on the light blue field are shown the contents of the cell in binary, hex, and as an ASCII character. The contents are shown as an ASCII character only if they are an alphanumeric or punctuation character. If the contents are a graphics or control character, it is not shown.

To understand exactly what you're seeing in the 64K memory cells of the Commodore 64 requires a memory map, such as the one found in the *Programmer's Reference Guide.* To get you started, however, here are a few of the more interesting memory locations on a C-64:

$00A0

Locations $00A0, $00A1, and $00A2 form the jiffy clock. The byte at $00A2 is incremented every 1/60 second (the 60 Hz AC power line provides the reference). Each time $00A2 reaches $FF, the byte at $00A1 is incremented, and each time $00A1 reaches $FF, the byte at $00A0 is incremented. You can't see every single change in the contents of $00A2 because while $00A2 is changing 60 times per second, *Scanner* is only updating $00A2 at the rate of 15 times per second.

$00C5

This location contains a code (not ASCII) indicating what key, if any, is being pressed. Pressing the 0 key, for instance, will produce a code of $23. A code of $40 means no key is being pressed.

$00C6

The byte at this location shows how many characters are currently stored in the keyboard buffer, up to a maximum of 10 ($0A). When you press f7 to enter a new address, this byte is automatically set by *Scanner* to $00.

$028D

This location contains $01 if the SHIFT key is pressed; $02 if the COMMODORE key is pressed; and $04 if the CTRL key is pressed.

$0400

Locations $0400-$07E7 are the screen RAM. They contain a coded representation of the current screen display. You can examine the screen codes that form *Scanner's* screen display by scanning through this area of memory.

$0801

Location $0801 is the beginning of the BASIC program area. With *Scanner,* you can examine the tokens and ASCII characters that make up a BASIC program. Load or type in your BASIC program, then load and run *Scanner* and scan through this area of memory.

$A004

Locations $A004-$A00C are part of the BASIC ROM. These locations contain the ASCII message CBMBASIC0.

$D808

Locations $D808-$D80C are part of color RAM. Only the low nibble (lower four bits) of these locations is significant, since a four-bit RAM chip is used for color RAM. The low nibble of all five of these locations will normally be $0, because the color of the graphics characters that form the top of cell $D80C is black, and the code for black is 0. Pressing f7 changes all the low nibbles to $C, because the color of the graphics characters that form the panel is medium gray, and the code for medium gray is 12 ($C). □

SEE PROGRAM LISTING ON PAGE 91

Reader Service No. 130

PRO-TECH-TRONICS

SOFTWARE

LIMITED SUPPLY (NO DEALERS)

ALL SIX PIECES OF SOFTWARE

1. English II
2. Math II
3. Science II
4. Triad
5. Pilot
6. Fish-Metic

$39

FREE SHIPPING!

SIMULATORS

GUNSHIP — F-15 JET

$28 ea.

PRO-SPORTS

GOLF LEADERBOARD — WRESTLING TAG-TEAM

$28 ea.

WORD PROCESSING

WORDWRITER WITH SPELL CHECK — TERM PAPER WRITER

$38 ea.

ACCESSORIES

POWER SUPPLIES
$29 C-64
$69 C-128

SURGE PROTECTOR $19
Protect your computer from lightning

DATA SET $28
For 64, PLUS 4

$58 TILT AND SWIVEL POWER CENTER

MOUSE $34.95
- Pull down menu
- Load directory
- Scratch
- Browse File
- Adjustable File

PEN LIGHT $89
"Create Flexdraw Computer Graphs"

FAST LOAD CARTRIDGE $29
Load your programs 5 time faster!

VOICE MASTER 64/128 $79
Command you computer by voice

JOY STICK $9.95
"with suction cups"
C-64, Plus 4, 128

MONITOR GLARE GUARD $19

RS-232 $48

FLOPPY DISK FILER $9.95

FREE TRIAL PERIOD FOR ALL ACCESSORIES

CALL TODAY
OPEN 7 DAYS A WEEK & EVENINGS

1-800-345-5080

Reader Service No. 130

Ahoy! DISK

Why type in the listings in this month's *Ahoy!* when we've done it for you? All the programs in this issue are available on the current *Ahoy! Disk* for $8.95. Isn't your time worth more than that?

Use the coupon at the bottom of this page to order disks for individual months, a disk subscription, or the special anthology disks described below. (You can also subscribe to the *Ahoy! Disk Magazine*— disk and magazine packaged together at a special reduced rate. See the card bound between pages 66 and 67.)

BEST OF '84
Address Book (3/84)
Air Assault (5/84)
Alpiner (6/84)
Checklist (7/84)
Construction Co. (12/84)
DOS (9/84)
Emerald Elephant (10/84)
I Got Biorhythm (5/84)
Lawn Job (10/84)
Lunar Lander (4/84)
Math Master (5/84)
Micro Minder (8/84)
Multi Draw 64 (1/84)
Name That Star (4/84)
Post Time (6/84)
Salvage Diver (9/84)
Space Lanes (VIC) (3/84)
Space Lanes (64) (3/84)
Space Patrol (12/84)
VIC 40 (10/84)

BEST OF '85
Ahoy! Dock (9/85)
Auto Line Numbers (3/85)
Directory Manipulator (12/85)
File Lock (9/85)
Fontasia (2/85)
Futurewar! (2/85)
Gameloader (12/85)
Home Budget (3/85)
Jewel Quest (12/85)
Lightning Loader (11/85)
Midprint (1/85)
Music Tutor (1/85)
Salmon Run (3/85)
Screen Dump (8/85)
Telelink (2/85)
Tile Time (5/85)

BEST OF '86
Disk Cataloger (8/86)
Dogcatcher (9/86)
Fidgits' Music School (7/86)
Flap! (8/86)
Guardian (11/86)
Head to Head (2/86)
Highlight (8/86)
JCALC (4/86)
Meteor Run (8/86)
Mine Canyon (9/86)
Mountaineer Mack (10/86)
Star Strike (6/86)
Step On It (8/86)
Syntax Patrol (9/86)
Teleporter (11/86)
Tenpins (11/86)
1541 Alignment Check (2/86)

BEST UTILITIES
Auto Append (7/85)
BASIC Trace (4/85)
Base Conversions (9/84)
Booter (4/85)
Commodore Font Editor (5/85)
DOS (9/84)
Fastnew (9/85)
Formatter (8/85)
Lucky Lottery (7/85)
Maestro (10/85)
Micro Minder (8/84)
Printat (7/85)
Screen Bit Dump (6/85)
Screen Magic (6/85)
Start & End File Address (9/85)
Two Column Directory (5/85)
Windows (8/85)
1541 Cleaning Utility (10/85)

BEST UTILITIES II
Alarm Clock (1/86)
BASIC Relocator (1/86)
Bigprint (5/86)
Chrono-Wedge (4/86)
Disk Checkup (10/86)
Failsafe (5/86)
Long Lines (11/86)
Programmer's Aid (3/86)
Quick Change (10/86)
Reversed Remarks (7/86)
Screen Sleuth (7/86)
Screen Window (4/86)
Speech64 (9/86)
The Editor (12/86)
Trackdown (6/86)
Very-Able (3/86)
128 to 64 Autoboot (10/86)

BEST GAMES
Alpiner (6/84)
Construction Co. (12/84)
Disintegrator (8/85)
Dragon Type (9/85)
Duck Shoot (6/85)
Emerald Elephant (10/84)
Gators N Snakes (8/85)
Gravinauts (10/85)
Lawn Job (10/84)
Mystery at Mycroft Mews (10/85)
Name That Star (4/84)
Post Time (6/84)
Salvage Diver (9/84)
Script Analysis (7/85)
Space Patrol (12/84)
Tile Time (5/85)
Wizard of Im (7/85)

BEST GAMES II
Actionauts (4/86)
Air Rescue (4/86)
Brickbusters (3/86)
Crabfight (2/86)
English Darts (5/86)
Got a Match? (6/86)
Haunted Castle (1/86)
Head On (7/86)
Knockout (1/86)
Micro City (12/86)
Minotaur Maze (12/86)
Penguins (10/86)
Ski Folly (5/86)
Star Search (5/86)
Swoop (4/86)
The Last Ninja (9/86)

BEST C-128
Ahoy!Term (3/86)
Dark Fortress (1/87)
Dual Dump (9/86)
Extended Background Mode (4/87)
Infoflow (4/87)
List Formatter (4/87)
More BASIC 128 (2/87)
Nebergall Run (7/86)
Planet Duel (4/87)
Queen's and Bishop's Tours (6/86)
Rescue 128 (3/87)
The Artist (12/86)
Variable Manager (9/86)
Wordcount (7/86)
128 to 64 Autoboot (10/86)

We've collected Ahoy!'s best programs onto the Anthology Disks described here. Please note that many of these programs will be unusable without the documentation printed in the issues of Ahoy! listed in parentheses.

SINGLE ISSUE AND SUBSCRIPTION PRICES

(Postage and Handling Included)	U.S.A.	CANADA	ELSEWHERE
❑ August '87 *Ahoy!* Disk	$ 8.95	$ 12.95	$ 13.95
❑ 12-Month Disk Subscription	$ 79.95	$ 99.95	$124.95
❑ 24-Month Disk Subscription	$149.95	$179.95	–

BACK ISSUES $8.95

In Canada add $2.00 per issue; outside US and Canada add $4.00 per issue.

❑ ____	$8.95	❑ ____	$8.95
❑ ____	$8.95	❑ ____	$8.95
❑ ____	$8.95	❑ ____	$8.95
❑ ____	$8.95	❑ ____	$8.95

ANTHOLOGY DISKS $10.95 EACH

❑ Best of '84 ❑ Best Utilities
❑ Best of '85 ❑ Best Utilities II
❑ Best of '86 ❑ Best Games
❑ Best of C-128 ❑ Best Games II

Send coupon or facsimile to: *Ahoy!* Program Disk
Ion International Inc.
45 West 34th Street, Suite 500
New York, NY 10001

MC and VISA Accepted Call 718-979-9330

NAME ____________________

ADDRESS ____________________

CITY ______________ STATE ________ ZIP ________

Make check or money order payable to **Double L Software**. N.Y. State Residents must add 8¼% sales tax.



BATTER UP!

For the C-64

By James C. Hilty



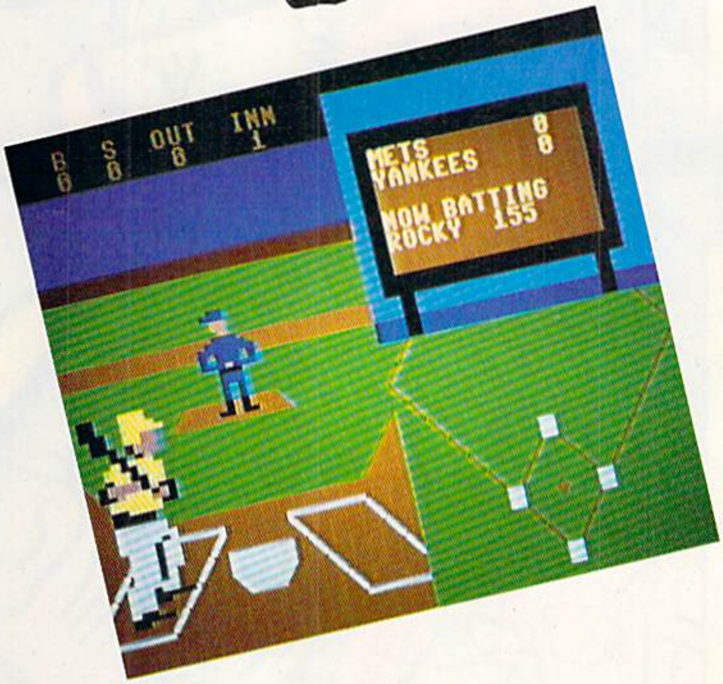


PHOTO: MICHAEL R. DAVILA

Popcorn, peanuts, hot dogs – the crack of the bat and the roar of the crowd – a runner heading for home in a race against an outfielder's accurate arm – now that warm weather is here the main topic of conversation is the national pastime, baseball.

Batter Up! is a one-player baseball simulation for the C-64 which will sharpen your batting eye against a clever computerized pitcher. The game features a split-screen playing field, joystick control, and an optional printing routine for making hard copies of lineups and batting averages. The game is written entirely in BASIC with numerous REM statements, so you can see what's going on with the program.

GETTING STARTED

Type in and save a copy of the program, then RUN it. The screen will prompt you for the name of your team. Next, the screen will ask you for a name for the computer team. Pressing the return key at either prompt will name the team, with the default name Pittsburgh for you and New York for the computer. The screen will then show you your batting order with the player's batting average next to his name. The batting averages are different each time the game is played. You then have the option of making a hard copy of the lineup and averages (on a 1525 or equivalent printer).

The playing field is drawn next. A split screen is used, with a batter on the left and an overhead view of the stadium on the right. The effect is that of watching a game on TV with one camera behind the catcher and another camera on the roof of the stadium. You are the visiting team and thus bat first. A bell will sound when the pitcher is ready to throw a pitch. The pitcher will go into his windup and throw the ball toward home plate. You have the choice of either taking the pitch or swinging at the ball by pressing the fire button on a joystick in Port 2.

The computer pitcher throws five different pitches. The ones outside the strike zone cannot be hit. When you hit the ball any number of things can happen, just as in real baseball. The player's batting average has a lot to do with it. For example, a player batting .340 has a better chance of hitting a home run than a player batting .180, but the .180 hitter will still occasionally drive one out of the park.

The game plays just like regular baseball, with one exception – if a player is walked, then any other runner on the basepaths moves up one base. Thus, a walk is the same as a single. The balls, strikes, outs, and inning are displayed on a small scoreboard on the left of the screen. The score, batter, and batting average are displayed on the large scoreboard on the right of the screen. The runners currently on the bases are represented by X's on the bases on which they are standing. The result of your hitting the ball is displayed on the right of the screen.

After your team makes three outs, the computer's results for that inning are displayed and it is your turn to bat again in the top of the next inning. The computer may score from 0 to 3 runs in each inning. Since the computer is the home team, the computer gets "last bat." After nine innings, the game is over.

STRATEGY

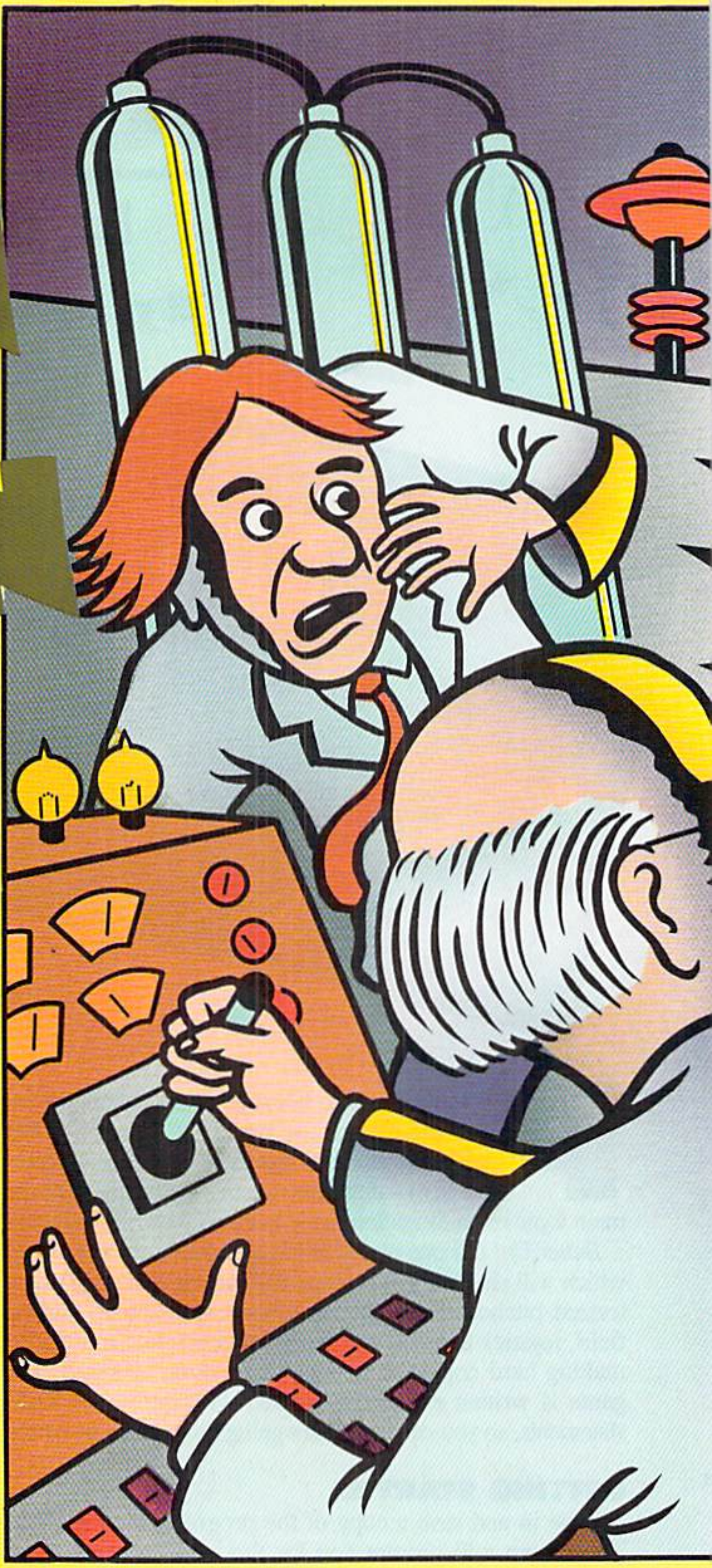
When first playing the game you should take some pitches so that you can see what the pitcher is capable of throwing. After a while you will be able to differentiate between the various pitches and will be able to see a strike coming.

Sometimes you will have a powerhouse team with high averages, and will probably want to swing away. In other games you might have a team with low averages, and you may want to take more pitches. Usually, you will have a lineup with a mixture of batting averages and you will want to manage accordingly.

The game moves along fairly fast; you will have to make quick decisions. In this game you have to be the batter and the manager. Now, the only remaining question is this – Are the Mets going to repeat, or are the Pirates going all the way? □ *SEE PROGRAM LISTING ON PAGE 80*

RUPERT REPORT

GREEN SCREEN BLUES



Using Interrupts for Electronic Screen Swapping

BY DALE RUPERT

Why should anyone in this age of electronic remote controls need to switch the video monitor from 40 column to 80 column mode by hand? That is one of the questions I have asked myself since I started using the Commodore 128.

The obvious answer is that we shouldn't. This month we will present a hardware/software project that eliminates the arduous burden of manually switching a "green screen" video monitor from one mode to the other. We will let the computer control an electronic relay to do the switching.

For those of you with dual-mode Red-Green-Blue (RGB) monitors such as the 1902, this hardware will not solve your problem. Perhaps the ideas along with the software presented here can be adapted for your system. In the meantime, you must still reach up and press that switch on your monitor.

Those of you who don't have a green-screen monitor and are not interested in controlling a relay with your computer may at least find the discussions of programming with interrupts useful.

We will set up an interrupt routine that 1) remains active even after a RUN STOP/RESTORE warm boot, and 2) can coexist with other such routines. These are not trivial requirements, but they are not difficult to achieve.

GREEN SCREENS

If you use your C-128 for serious work (as opposed to serious game-playing), you should be taking advantage of the 80 column mode. The least expensive and in some respects the best way to use 80 column mode is with a "com-

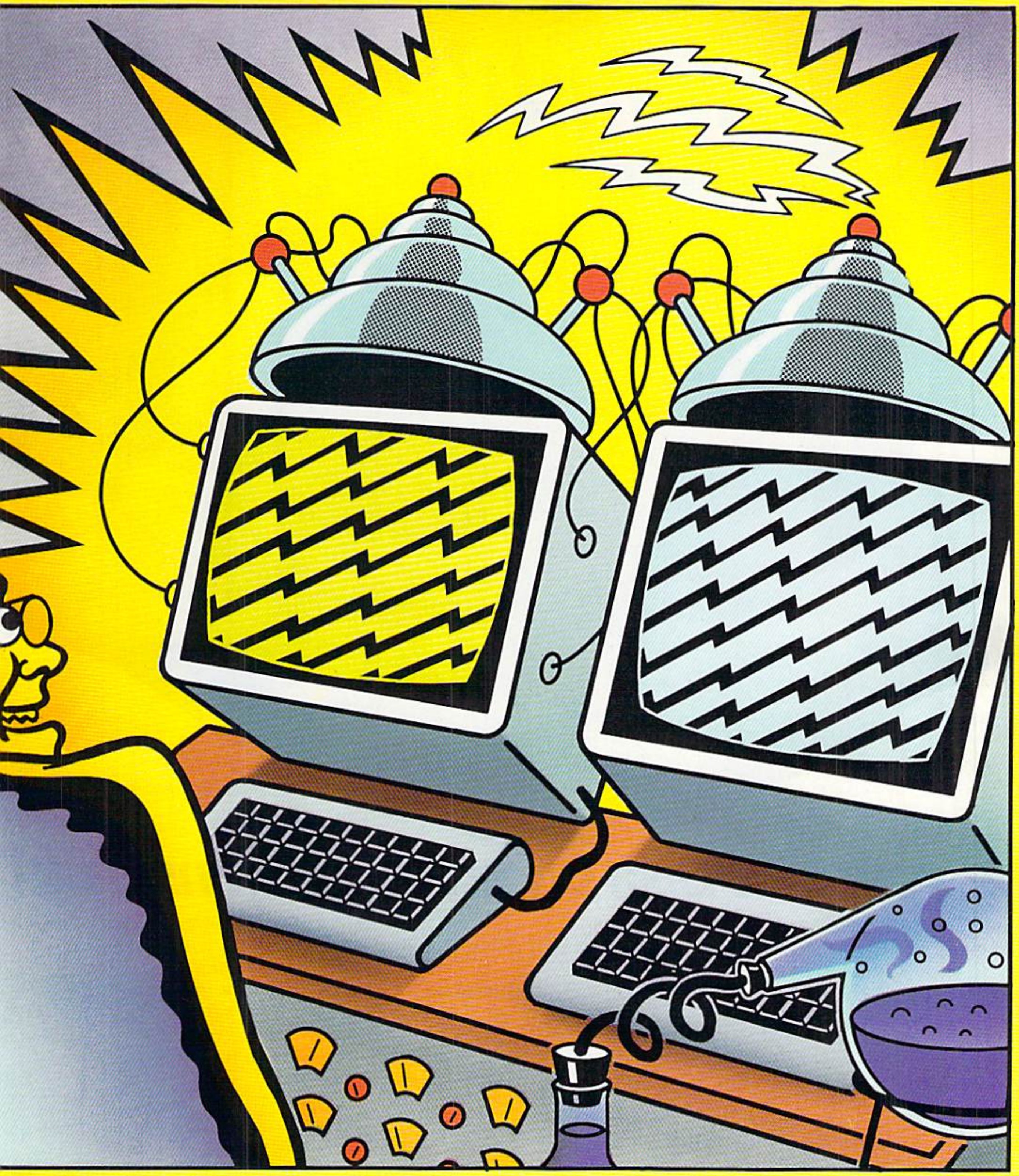


ILLUSTRATION: TOM CUSHWA

posite" monochrome video monitor, usually called a "green screen."

These are monochrome displays with green or amber phosphors, and usually with very high bandwidth and resolution. The bandwidth is a measure of the sharpness of an image on the screen. A standard television set may have a bandwidth around three or four megahertz. It is not uncommon for a green screen monitor to have a bandwidth of twelve or fifteen megahertz. The higher the bandwidth, the sharper the image.

The electron beam zooming across the screen is turned on and off at appropriate times causing some phosphors to glow and others to blank, thus generating the video image. Low bandwidth circuitry simply cannot turn the electron beam on and off quickly enough to show the individual pixels of 80 characters per line. You must squint and use a lot of imagination to read 80 column text on a television set.

In addition to TV sets and green screens, there are RGBI



Photo by: **Bobbi**
Grenada, California

*Trademark of Commodore Business Machines

Reader Service No. 129

Give your 1541* Disk Drive a brain transplant!

Simply plug the **STARDOS ROM** Chip into the drive unit and insert the **STARDOS** cartridge. That's all you need to do to complete a transplant procedure that will give your Disk Drive and Computer higher intelligence.

Other Quick Loaders and Fast Loaders try to 'Patch' the DOS of the Disk Drive. This does not significantly speed up much of today's Software and sometimes it fails to load altogether.

But, **STARDOS** electronically replaces the entire DOS and KERNAL. This means **STARDOS** will speed-up finding records in your Database, any changes you wish to make to your Spreadsheet and playing your favorite adventure or other Disk Intensive work with your C-64*.

The **STARDOS** transplant offers:

Speed............. *for 'All Types' of Disk Access at an incredible 600%-1000% faster!*

Reliability....... *end overheating and "SAVE @ O:" woes!*

Convenience.. *with single key load, SAVE, and Directory Etc. Built in File and Disk Copiers!*

Power............. *full Word Processor, M.L. Monitor and Sector Editor, Screen Dumps to Printer!*

Compatibility. *STARDOS works with virtually all C-64* Software and Hardware!*

Guarantee....... *120 day repair or replacement warranty. Your satisfaction guaranteed or your money is refunded!*

Help your C-64* live up to its full potential...
Transplant with **STARDOS!**

Personalize your C-64* for a $10 *(non-refundable)* charge. We will include a 21 character power-on message with your favorite screen, border and text colors. ***Example:***

MIKE'S COMMODORE-64

Background BLACK Border GREEN Text RED

Write or phone for additional information.

STARPOINT SOFTWARE

6013 MACKS GULCH RD. GAZELLE, CA 96034-9412 (916) 435-2371

When ordering by mail:
$64.95 + 3.00 shipping
$64.95 + 4.00 COD orders

Shipping out of USA $6.00
California residents add 6% sales tax
VISA or Mastercard accepted

Please allow 4-6 weeks for delivery

(red-green-blue-intensity) monitors, usually called RGB monitors. To get high-resolution text and multiple colors, you need one of these "direct input" monitors. The RGB monitor is much more costly than a green screen. Text on a $600 RGB monitor will be almost, but not quite, as sharp as text on a $100 green screen. Text on a $300 RGB monitor is not nearly as clear as on a green screen.

The C-128 takes very limited advantage of its RGB circuitry. (Ever try to draw red, green, or blue circles in 80 column mode?) This may sound like heresy, but it seems a waste to attach an expensive RGB monitor to the C-128. Why? Because the C-128's RGB output (80 column mode in BASIC) only supports text, and a green screen is better than an RGB monitor for text.

What's the best solution? If you use your C-128 computer primarily for programming and word processing, get a green screen. Period. If you also want to play games or generate color graphics, get a separate color composite (not RGB) monitor, such as the 1701 or 1702, or use a television set. Another possibility is to use one of a new breed called a "television/monitor" which allows you to bypass the RF (radio frequency) circuitry. (In general, the RF section limits the sharpness of the image.)

The C-128 allows you to select either the 40-column output or the 80-column output. With this month's project, the computer will automatically route the selected output to the green screen.

THE HARDWARE

Building this project is a straightforward task. Refer to the schematic diagram on page 36. The Parts List on this page includes Radio Shack numbers in parentheses. There are many other sources for these parts. If you use a relay different from the one listed here, you may need a different value of resistor. We will discuss that later.

You may use any method to assemble the project. Probably the neatest and easiest is to use a pre-drilled and -etched universal printed circuit board such as the one with the box in the parts list.

Refer to the Assembly Tips on page 40 to help with the construction. Be sure to figure out how to fit all the cables and components in the box before you begin soldering things together. Also determine the placement of the parts on the printed circuit board, if you use one.

The relay is an electrically controlled switch attached to the cassette motor driver. When the computer is first turned on, there is no power supplied to the relay coil, and the relay switch is in the Normally Closed position. This brings the output from the 40-column Video port to the video monitor. If the C-128 is used in C-64 mode, the relay is not powered, and the 40-column output is selected.

To bring the 80-column signal from the RGBI port to the monitor, power must be applied to the relay coil. It becomes an electromagnet which pulls the switch from its Normally Closed position to the Normally Open position. You can hear a click as the relay switch changes positions.

Power is brought to the relay coil from the motor driver circuit of the cassette port. Software controls the state of that circuit. Software that normally turns on the cassette motor now energizes the relay.

PARTS LIST

Relay (275-240)*: SPDT, 5 VDC, 90 mA
Resistor (271-005)*: 22 ohms, ¼ or ½ watt
Diode (276-1122): 1N914 or equivalent switching diode
Phono plug and cable (42-2371): 6' coax with one plug
DIN plug (274-003 or 274-020): 5 or 6 pin male
Sub-D plug (276-1537): 9 pin male
Sub-D hood (276-1539): to fit 9 pin plug
Card-edge socket* (part of 276-1551): 0.156", 12 pins

Optional:
Box and circuit board (270-283): enclosure and universal etched and drilled printed circuit board

(Radio Shack part numbers are shown in parentheses)
*Refer to the article for details

The diode is an important part of the circuit. Be sure it is connected properly. Do not use the circuit without it! The relay generates a voltage surge whenever it is deenergized. The diode keeps the surge from damaging the driving transistor inside the computer.

If you use a relay different from the one listed, you may need a different value of resistor. There are two types of ratings associated with a relay: the coil rating and the contact rating. The contact rating tells how much current can

be safety switched by the relay (typically 1 or 2 amps at 125 volts AC). The video signals we are switching are very small compared to the maximum allowed contact ratings of most relays. We are only concerned with the coil rating.

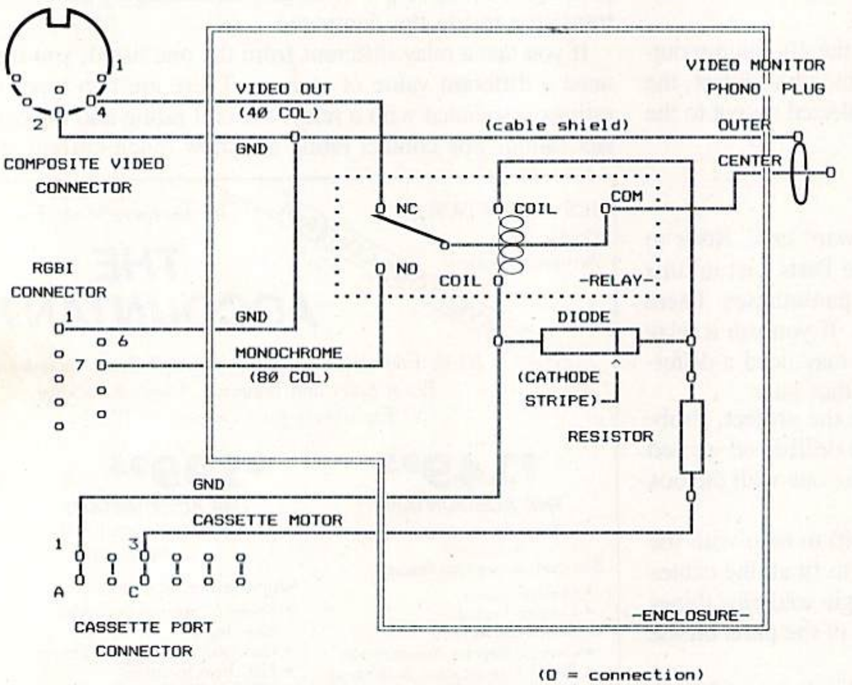
The coil rating tells the nominal voltage and current required to make the relay coil operate. The relay in the parts list has a coil rating of 5 volts DC, 55 ohms resistance, and 90 milliamps of current. Actually you need only any two of these numbers. The third is found from Ohm's law: V = I * R where V is voltage, I is current in amps, and R is resistance in ohms. Verify that 5 volts approximately equals 0.090 amps (90 milliamps) times 55 ohms.

The output of the cassette motor circuit is nominally 6 volts, though it could be close to 7 volts. Use a relay with a coil rating of 6 volts or less. For a conservative design, we'll assume the motor output voltage is 7 volts. With a nominal 5-volt coil relay, the remaining 2 volts from the cassette port must be dropped across a resistor.

The value of the resistor is calculated from Ohm's law since the voltage across it is known (7 volts minus the relay coil voltage) and the current through it is the same as the current through the relay coil.

Assume you use a relay with a nominal 6 volt coil which needs 100 milliamps to turn on. There will be 1 volt across the resistor (7 − 6 = 1). The current through the resistor will be 0.1 amp (100 milliamps). Therefore the value of the resistor must be 10 ohms (1 volt / 0.1 amp = 10 ohms) from Ohm's law. Only certain values of resistors are available. If your calculated value is not available, choose the next higher value.

A relay will not reliably turn on if the resistor value is too large, since there will not be enough current through its coil. If the value of the resistor is too small, too much current will flow, possibly damaging the relay and the circuitry inside the computer.



SCREEN SWAPPER SCHEMATIC

The only other hardware concern is the power rating of the resistor. Resistor power ratings range from fractions of a watt to several watts. To calculate the minimum power rating of the resistor, multiply the resistance times the square of the current: P = I↑2 * R.

For the resistor in the parts list, current is 0.090 amp and resistance is 22 ohms, so its power consumption is 0.18 watt (0.09 ↑ 2 * 22). We must use a resistor with a power rating at least this great, so ¼ watt or larger is the proper choice. A ⅛ watt (0.125 watt) or smaller resistor cannot dissipate heat rapidly enough for this circuit and may eventually burn out.

Now let's look at the software to control this hardware.

THE SOFTWARE

We want the software to have the following characteristics:

1. It must continuously check the display mode of the computer (40 or 80 columns).
2. It must turn the cassette motor output on when the computer is in 80-column mode and must turn the motor output off when in 40-column mode.
3. It must be loaded once and remain active without interfering with other programs.
4. It should not be disabled with a RUN STOP/RESTORE warm boot.

The first and third requirements imply the use of the IRQ (Interrupt ReQuest) interrupt routine. The computer executes the IRQ routine every sixtieth of a second. The normal IRQ routine does such things as update the jiffy clock and scan the keyboard. It is possible to patch our program into the IRQ routine so that it will check the video mode and set the relay accordingly.

When an interrupt occurs, the computer calls the subroutine whose address is stored in RAM locations $314 and $315 (hex). The normal IRQ routine address is $FA65. We will put the address of our program into locations $314 and $315. When the interrupt occurs, the computer will then call our program. Our program will conclude by sending the computer on to the original IRQ routine. This is a process called "wedging." We will wedge our program into the normal interrupt sequence.

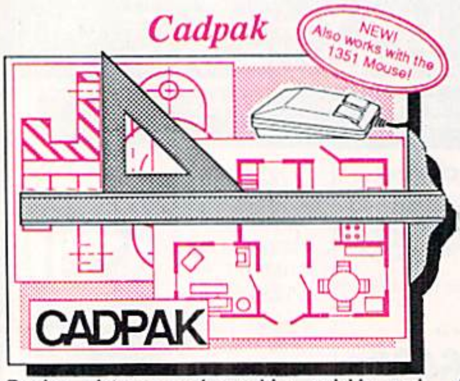
Requirement number four is the trickiest constraint. Normally the warm boot routine rewrites the original IRQ address ($FA65) into locations $314 and $315, breaking the link to our program. After that our program would no longer be executed by the interrupt routine.

A standard solution is to tell the user to enter a SYS command after a warm boot. This would execute a simple machine language routine to replace the vector at $314/$315 with the address of our program once again.

Reader Service No. 127

HOT SELLERS

Cadpak



Design pictures and graphics quickly and precisely. Unlike other drawing programs, you can produce exact scaled output on your printer. Design in the units of your drawing problem (feet, miles, meters, etc.) and send hardcopy to most printers. Uses either the keyboard, lightpen or 1351 mouse. Two separate work screens—transfer artwork from one screen to the other. Place text in four sizes anywhere in the picture—three extra fonts included: Old English, 3-D and Tech. "Try Again" allows you to undo mistakes. Draw solid or dashed lines, circles, ellipses at any angle, rays and boxes. Design fill patterns, fonts and objects. ***Cadpak*** is the full-featured design and graphics package for your computer.

for C-64 **$39.95** for C-128 **$59.95**

Super Pascal

Your complete system for developing applications in Pascal. A complete implementation of standard Pascal (Jensen and Wirth). C-64 version has a high-speed DOS (3X) for quick and efficient use. The extensive editor (source included) contains added features: append, search and replace. Includes assembler for any of your



machine code requirements. Used in hundreds of schools to teach programming and also used for serious development projects. But it can be used for more than just learning Pascal, use it for serious programming. With complete graphic library (source included) in machine language for super-fast execution. Want to learn Pascal or develop software using the best tool? ***Super Pascal*** is your first choice.

for C-64 **$59.95** for C-128 **$59.95**

BASIC Compiler

Now anyone can speed up their BASIC programs by 3 to 35 times! Basic-64 and Basic-128 easily convert your programs into fast machine language or speedcode (takes up less space yet protects your programs from prying eyes) or a mixture of both. You can even compile programs written with extentions—Simon's Basic, VICTREE, BASIC 4.0, VideoBasic and others. When the compiler finds an error, it just doesn't stop, but continues to find any other errors as well. Supports overlays and has many other options. 128 version works in FAST mode and allows you to use all 128K of memory. If your program walks or crawls, give it the speed to *RUN!*

for C-64 **$39.95** for C-128 **$59.95**

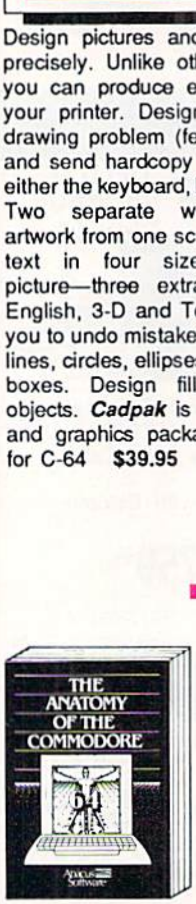
"...everything a good compiler should be...easy to use...efficient...offers a good range of optional features...excellent documentation...inexpensive."
Tom Benford, Commodore Magazine

BASIC COMPILER



Give your BASIC programs a boost!

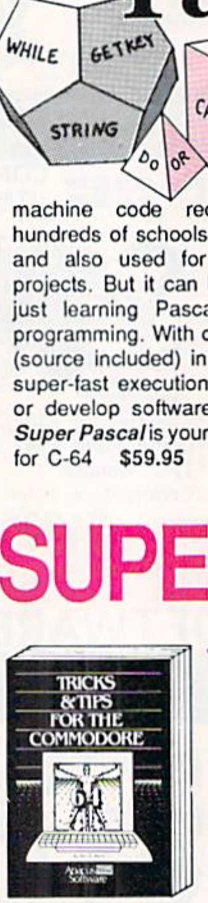
. . . and SUPER BOOKS!



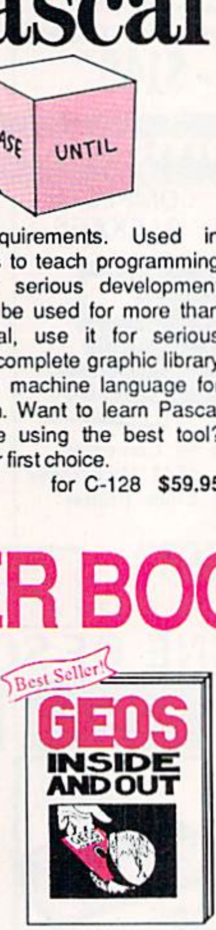
Anatomy of the C-64
Insider's guide to '64 internals. Graphics, sound, I/O, kernal, memory maps, and much more. Complete commented ROM listings. **300pp** $19.95



Anatomy of the 1541 Drive
Best handbook on this drive, explains all. Filled with many examples programs, utilities. Fully commented 1541 ROM listings. **500pp** $19.95



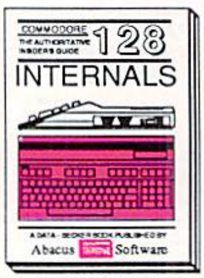
Tricks & Tips for the C-64
Collection of easy-to-use techniques: advanced graphics, improved data input, CP/M, enhanced BASIC, data handling and more. **275pp** $19.95



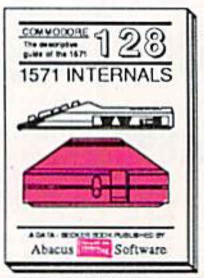
GEOS Inside and Out
Detailed info on GEOS. Add your own applications to GEOS. Edit icons. Constant display clock. Single-step through memory. $19.95



GEOS Tricks and Tips
Collection of helpful techniques for all GEOS users. Includes font editor, machine language monitor, quick backup, more. $19.95



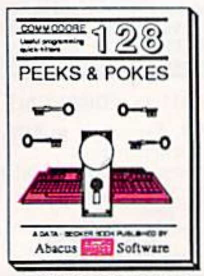
C-128 INTERNALS
Important C-128 information. Covers graphic chips, MMU, I/O, 80 column graphics and fully commented ROM listings, more. 500pp $19.95



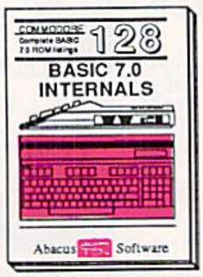
1571 INTERNALS
Essential reference. Internal drive functions. Explains various disk and file formats. Fully-commented ROM listings. 450pp $19.95



C-128 TRICKS & TIPS
Fascinating and practical info on the C-128. 80-col hires graphics, bank switching. 300 pages of useful information for everyone. $19.95



C-128 PEEKS & POKES
Dozens of programming quick-hitters, techniques on the operating system, stacks, zero page, pointers, and BASIC. 240pp $16.95



C-128 BASIC 7.0 Internals
Get all the inside info on BASIC 7.0. This exhaustive handbook is complete with fully commented BASIC 7.0 ROM listings. $24.95

Call now for the name of your nearest dealer. Or order direct with your credit card by calling 616/241-5510. Add $4.00 per order for S&H. Foreign add $12.00 per item.

Other books and software also available. Call or write for your free catalog. Dealers inquires welcome–2000 nationwide.

Abacus
P.O. Box 7219
Dept. H8
Grand Rapids, MI 49510
Telex 709-101 • Fax 616/241-5021
Phone 616/241-5510

We don't want a standard solution. We want the warm boot routine itself to put our program's address into the IRQ pointer at $314/$315. The secret to doing this is derived from the book *Commodore 128 Programming Secrets* by William Wiese, Jr. (Osborne McGraw-Hill).

During the warm start sequence (but after it has rewritten the IRQ vector), the computer jumps to the address contained in the SYSTEM_VECTOR in RAM locations $A00/$A01. Normally this is the BASIC warm start routine at $4003. All we need to do is replace the $4003 at $A00/$A01 with the address of our own initialization routine.

Our initialization routine will restore the IRQ vector to point to our main program. It will also check the 40/80 column mode and set the relay to select the proper video output. Then our initialization routine will continue with the normal warm start sequence at $4003.

PEACEFUL COEXISTENCE

One final but very important consideration is that our IRQ wedge should be able to coexist with other programs which use the IRQ vector or the SYSTEM_VECTOR. How many times have you loaded a utility only to find that another utility in memory now no longer works? There is no reason for such a thing to happen if both utilities are written properly and if they are not stored in the same memory locations.

The simple solution is to read and save the vector you are replacing rather than merely to assume the default vector was there. An example will clarify this. Locations $314/$315 normally contain the address $FA65. Assume we have two programs which are both to be wedged into the IRQ routine. P1 resides at address $1400 and P2 at $1500.

If P1 were an inconsiderate program, it would simply put its starting address at $1400 into $314/$315 and its last statement would be JMP $FA65. If P2 were also an inconsiderate program and were loaded next, it would put $1500 into $314/$315. It thereby overwrites the $1400, and P1 is no longer called by the IRQ routine. If you reinitialize P1, then P2 no longer works.

Here is how the two programs should be written. P1 reads the value in $314/$315 and saves it in a variable called P1VEC, for example. Assume the value it read is the default value $FA65, so P1VEC equals $FA65.

P1 then stores its own main routine starting address ($1400) at $314/$315 so the interrupt routine will call P1. The last statement in P1's main routine would be JMP (P1VEC), that is, "jump to the location whose address is stored in P1VEC." This is equivalent to JMP $FA65, just as before. So far, so good.

Now load and initialize P2 the same way. P2's initialization routine reads the value in $314/$315 and saves it in variable P2VEC. This value is not the default value $FA65. Instead it is the starting address put there by P1, so P2VEC equals $1400.

P2 wedges its main starting address ($1500) into $314/$315. The last instruction in P2's main routine is JMP (P2VEC).

ASSEMBLY TIPS

1. Cut off two eight-inch lengths from the free end of the coaxial phono plug cable. Strip the outer casing back about ½ inch from both ends of each of the two pieces and at the severed end of the original cable. Be careful not to cut very much of the shield wires. Strip the inner insulation back about ¼ inch from each end. Twist each of the braided shields.

2. Solder the inner wire of one 8-inch piece of cable to pin 7 of the 9-pin D connector. Solder the twisted braid of that piece to pin 1 of the D connector. Label the other end of that wire "NO". (See Step 4.) Solder the inner wire of the other 8-inch piece to pin 4 of the DIN plug. Solder its braid to pin 2 of the DIN plug. Label the other end of that wire "NC".

3. Twist and solder the three free braids together (two from the 8-inch cables in step 2 and one from the original phono plug cable). You may want to tie-wrap the three cables together.

4. The inner wires of the three cables are attached to appropriate pins on the relay: the inner wire of the 9-pin D cable goes to the Normally Open (NO) terminal, the 6-pin DIN cable inner wire goes to the Normally Closed (NC) terminal, and the phono plug inner wire goes to the Common (COM) terminal of the relay. If there is more than one Common terminal, it may go to either one.

5. The cathode end of the diode goes to one side of the relay coil and also to one end of the resistor. (The cathode end of the diode is marked with a stripe.)

6. If you use the 44-pin card-edge connector (0.156 inch spacing) in the parts list, you must cut it to make a 12-pin connector. Just use a hacksaw and cut it through pin 7. (The longer piece of the connector can be cut again through pin 10 to make a 24-pin connector which fits the user port.)

7. Cut two pieces of wire, one red and the other black, about eight inches long. Solder the other end of the diode (its anode) and the black wire to the other side of the relay coil. Solder the red wire to the open end of the resistor. Solder the other end of the red wire to pins C/3 of the 12-pin cassette port card-edge connector. Solder the black wire to pins A/1 of the 12-pin card-edge connector.

That's all there is to it. The 9-pin D connector goes to the RGBI port, the circular DIN connector goes to the Video port, the 12-pin card-edge connector goes to the Cassette port on the C-128. The phono plug goes to the video monitor input.

What does this do? It causes execution to continue at address $1400, so now P1 will also be executed. All properly written IRQ wedges will be executed, but opposite to the order in which they were loaded.

Any number of IRQ wedges can be implemented this way. The only constraint is that the routine obviously must not occupy the same memory space. One may have to be relocated if they overlap.

The same concepts apply to programs which are to be wedged into the warm start routine through the SYSTEM_VECTOR at $A00/$A01. Let's put these principles to work.

Look at the assembler listing SCRNSWAP.ASM on page 83 for the following discussion. The program has three separate entry points, the Initialization (line 13), the warm boot Restart (line 39), and the Main routine (line 62). The Initialization is called from BASIC when the program is first loaded. The Restart is called after a RUN STOP/RESTORE reboot. The Main program is called every sixtieth of a second by the interrupt routine.

The initialization routine begins at address $1500. Variables IRQVEC and SYSVEC in lines 157 and 158 will store the values read from the IRQ pointer at $314/$315 and the SYSTEM_VECTOR pointer at $A00/$A01. The initialization will be executed by a SYS 5376 or SYS DEC("1500") statement in BASIC.

Lines 115 through 118 determine if this initialization has already been executed. If so, it is not repeated. If not, lines 120 through 122 read the IRQ vector from $314/$315 and save it in IRQVEC. Lines 124 through 127 save the SYSTEM_VECTOR in SYSVEC.

Lines 129 through 132 wedge the starting address RESTART of the warm boot portion of the program into the SYSTEM_VECTOR. CONFIG at line 142 is called to read and save the current video mode, initialize the relay, and wedge the MAIN routine into the IRQ vector. The initialization portion of the program is done, and it returns to BASIC in line 135.

Now the RUN STOP/RESTORE warm boot sequence will branch to our program beginning at address RESTART in line 140. The RESTART routine calls CONFIG beginning at line 143 to determine the video mode, to initialize the relay, and to wedge the MAIN portion of our program into the IRQ vector. Finally in line 141 the RESTART routine jumps to the vector it previously saved in SYSVEC.

The IRQ vector points to our MAIN program at line 162. This program is executed every sixtieth of a second. The program compares the previous video mode saved in variable PREVMD to the current mode. The current video mode is indicated by bit 7 of location $D7 (MODEFLG). If this bit is 1, 80 column mode is selected; 0 corresponds to 40 column mode. The current mode is saved in line 167. If the current mode is the same as the previous mode, the routine exits in line 171 to continue with the normal IRQ routines.

If the video mode was changed during the previous sixtieth of a second, line 169 branches to the RELAY routine at line 174. Lines 175 through 182 perform an initialization so that the motor output of the cassette port can be changed. Line 183 decides if the relay should be turned on for 80 column mode or off for 40 column mode.

If 40 column mode is selected, the relay is turned off. Also the 40 column display is turned on by setting bit 4 of the VIC control register at $D011. A zero is stored in the processor clock rate register to put the computer into SLOW mode.

If 80 column mode is selected, the relay is turned on. The speed of the computer is not changed. This allows the

Continued on page 60

ACTION IN OUTER SPACE

The Hottest Science Fiction Arcade Software

By Arnie Katz & Bill Kunkel

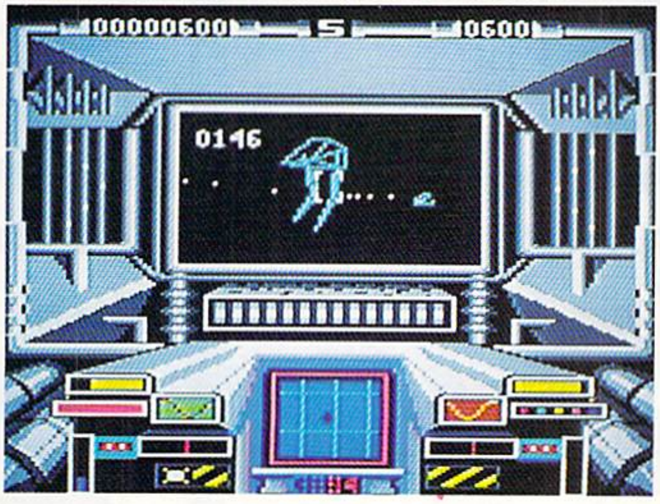
When the cares of the world hang heaviest, there's something invigorating about an unabashed shoot-em-up. Even an ardent adventure gamer can be coaxed away from parser-pounding, at least temporarily, by the speed, the sounds, and the excitement of a rousing science fiction blastathon.

The sedate *Pong* was the very first computer game for the masses, but it didn't take space action games long to blast into the public consciousness. A solitaire contest in which the player shot at a pesky flying saucer with a spaceship cannon was raking in quarters at amusement centers within a year after the *Pong* craze hit.

The years 1984-1986 weren't generous to fans of action science fiction software. The much-discussed slump in real-time computer games hit the science fiction category especially hard. The public had always closely identified science fiction themes with the suddenly unchic videogames, and computer funware publishers didn't want guilt by association.

It is axiomatic that when publishers stop buying a type of software, designers stop trying to create it. Authoring entertainment software is a career, after all, and it's hard to make a living out of unsold product.

In the U.S., talents once devoted to the zap and pow of outer space combat turned to hot game themes like sports simulations and adventures. Science fiction action disks, which once seemed as numerous as the stars in the Milky Way, had become as rare as Mr. Spock's smiles by summer 1986.



Starglider *is a largely original arcade action game. Vector style line graphics help enhance the sensation of moving at great speed, as well as free-falls and sudden stops. READER SERVICE NO. 119*

Help has come, not from across the firmament, but from across the Atlantic Ocean. Britain's Videogame Era had a more modest boom and a far less disastrous bust. No one told U.K. publishers, designers, and consumers that they should stop liking science fiction action programs, so they didn't. As a result, hundreds of SF-oriented joystick jamborees have made their debut in Great Britain in the last two years.

These new titles aren't just the same old reflex-testers, either. They reflect today's tastes in audiovisual effects and sophisticated play-mechanics. U.K. space battle programs not only look better than many of the science fiction classics, but most mix just enough strategy to keep play from becoming numbingly repetitive.

Jeremy San's *Starglider* (Firebird) is a rarity among entertainment software products: a largely original arcade-style shoot-out. This C-64 translation by Solid Image gives American computerists a chance to enjoy a program that's already been a smash in England.

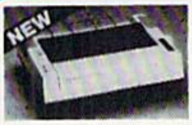
Starglider is an innovative first-person space combat contest. Although it visually resembles vector graphics games, and it shares some elements with *Battlezone* and *Stellar 7,* the total package is remarkably fresh.

The game begins when the player's AGAV (Airborne Ground Attack Vehicle) zooms into the fray at near-light speed. The AGAV is capable of outrunning anything your Alliance, Egron, or Aruloid foes can send after it, short of light-speed vehicles.



Reader Service No. 150

Panasonic Industrial Company



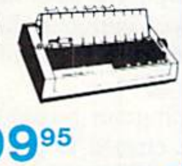
KX-P1080*i* $174.95



$229.95 KX-P1091*i*



KX-P1092*i* $299.95



KX-P1592 $399.95

SEIKOSHA



• w/2 ribbon purchase

SP180VC $139.95*

SP1000VC $149.95*

SEIKOSHA MP1300A1



300 CPS
10K Buffer
Parrellel + Seriel
50CPS N.L.Q

$349.95 Color Version $449.95

Avatex



1200 Baud...... $98
1200 H.C..... $120
2400 Baud.... $220

DISKS & ACCESS.

MOUSE PAD........... $6
CASIO CZ101........ $250

3½ DS/DD (30 lot)	$1.30
3½ SS/DD (30 lot)	$1.00
5¼ DS/DD (100 lot)	$ .30

3½ DISC CASE (30) $8
5¼ DISC CASE(75) $8
DISC HEAD CLEANERS $8

SPECIALS

Commodore Logo	$25
Accounts Rec.	$20
Silent Butler	$10
Xetec Interface	$CALL
Kwik Load	$5
Joy Sticks	from $5
Outlet Surg. Strip	$20

commodore 1520 COLOR Printer Plotter



commodore Educator 64



$149.95*

Completely Commodore Compatible

commodore SFD 1001 (1 Meg)



$179.95

4040 ...$ CALL
8250 ..$ CALL*

commodore Hard Drives 9060



$450*

Mini Chief/Data Chief ....$ CALL

commodore PET



4023 . 99.95*
8023 .. 150*
8032 .. 150*

*Completely Refurbished

AMIGA NEW RELEASES

X-rated Graphics Library	$ 15	Gizmoz 2.0	$ 42	Prowrite	$ 75	Earl Weaver Baseball	$CALL
Publisher 100	$120	Starglider	$ 27	Word Perfect	$240	Perfect sound	$ 48
Digipaint 2.0	$ 36	City Desk	$ 90	Laser Script	$ 27	Acquisition	$180
Faery Tale Adv.	$ 30	Galileo	$ 60	Surgeon	$ 36	Videoscape 3D	$120

ALL SOFTWARE AT LEAST 40% OFF

EA

Bard's Tale	$24.00
Chessmaster 2000	30.00
Lords of Conquest	21.00
Marble Madness	24.00
Pegasus	18.00
Moebius	24.00
Murder Party	21.00
Ogre	24.00
Ultima IV	36.00
Ultimate Wizard	24.00
Bard's Tale II	24.00
Auto Duel	30.00
Scrabble	21.00
America's Cup	21.00
Artic Fox	24.00

MINDSCAPE

Defender of the Crown	NEW $21.00
Bop and Wrestle	18.00
Fist	18.00
Super Star Hockey	NEW 21.00
High Roller	18.00
Indoor Sports	18.00
Infiltrator	18.00
Halley Project	18.00
Parallax	18.00
Uridium	18.00
Luscher Profile	18.00

ACCOLADE

Hard Ball	Special $17.00
Fight Night	18.00
Dambusters	18.00
Law of the West	18.00
PSI 5 Trading Comp.	18.00
Kill Until Dead	18.00
Ace of Aces	18.00
Comics	18.00

PRECISION

Superbase 128	$50.00
Superbase 64	40.00
Superscript 128	40.00
Superscript 64	40.00

50% OFF Berkeley Softworks

GEOS	Special $34.00
GEOS Fontpak 1	18.00
GEOS Desk Pack	21.00
Writer Work Shop	30.00
Geodex	24.00
GeoCalc/GeoFile	NEW 30.00
GeoCable	NEW 24.00

Abacus

GEOS In and Out	$9.00
Basic 128	$36.00
Basic 64	24.00
Cadpak 128	36.00
Cadpak 64	24.00
Chartpak 128	24.00
Chartpak 64	24.00
Cobol 64	24.00
Cobol 128	36.00
Super C Compiler 64/128	36.00
Super Pascal	36.00

Super Pack 64	Special $36.00
Super Pack 128	45.00
Pocket Writer 2/Filer/Planner	EACH 36.00
Super Pack 2 (128)	60.00
Pocket Writer 128/Filer/Planner	EACH 30.00
Pocket Writer 64/Filer/Planner	EACH 24.00

ACTIVISION

Portal	$24.00
Titanic	18.00
Labrynth	21.00
Aliens	21.00
Hacket	18.00
Hacker II	21.00
Little People	21.00
Game Maker	24.00
Music Studio	18.00

SPRINGBOARD

Certificate Maker	Special $28.00
Newsroom	$30.00
Clip Art 1	18.00
Clip Art 2	24.00

FIREBIRD

Knight Ork	NEW $24.00
Colossus IV	21.00
Elite	18.00
Frankie Goes Holly	21.00
Pawn	24.00
Talking Teacher	24.00
Guild of Thieves	NEW 24.00

UNISON WORLD

Art Gallery	$16.00
Art Gallery 2	16.00
Print Master	21.00

HI TECH

Card ware	$6.00
Heart ware	6.00
Party ware	6.00
Ware with all	9.00

Broderbund Software

Printshop	Special $25.00
Graphics Lib. EA	15.00
Carmen Sandiego	21.00
Karateka	18.00
Print shop comp.	21.00

TIMEWORKS

Data Mgr. 128	$36.00
Swiftcalc 128	36.00
Wordwriter 3	30.00
All business titles	36.00
Partner 64	36.00
Partner 128	42.00
Sylvia Porters	36.00

MicroProse

Gunship	Special $20.00
F-15	21.00
Acrojet	20.00
Kennedy Approach	21.00
Silent Service	24.00
Pirates	NEW 24.00
Top Gunner	18.00

Xetec, Inc

Font Master II 64	$30.00
Font Master 128	36.00

MicroLeague

Micro L. Baseball	$24.00
General Mgr.	24.00
Stat Disk	15.00
'86 Team Disk	12.00

SSI

Gemstone Healer	$18.00
Gettysburg	36.00
Kampfgruppe	36.00
Phantasie 2	24.00
Ring of Zelfin	24.00
Road War 2000	24.00
Shard of Spring	24.00
Wizard's Crown	24.00

EPYX

FastLoad	Special $22.50
Movie Monster	24.00
Vorpol Kit	21.00
World Karate Champ	18.00
Summer Games II	24.00
Super Cycle	24.00
World Games	24.00
Football	24.00
Multi-Plan	24.00
Winter Games	24.00
Sub Battle	NEW 24.00

LOGIC

Flight Sim II	Special $28.50
Scenery Disk	12.00
Jet	30.00
Football	24.00
Baseball	30.00

ACCESS Software Incorporated

MACH-128	$28.50
Leaderboard	24.00
MACH-5	24.00
World Class leaderboard	Special 22.50
10th Frame	24.00
Exec Tournament	12.00
Tournament 1	12.00
Triple Pak	12.00
Famous Course	12.00



Uridium: *for carnage connoisseurs.*
READER SERVICE NO. 120

The most exciting feature of the *Starglider* craft, however, is that it can save on fuel by gliding for great distances. To accomplish this, the pilot takes the AGAV to a decent altitude and kills the engines. This is not only an energy-saving strategy, but it is also quite useful in attack situations.

Starglider communicates the sensation of moving at great speed, as well as sudden stops and freefall, better than any previous flying and shooting program. The vector-style line graphics, which convert objects to crystalline, three-dimensional sculptures, were an excellent choice to enhance this perception. Enemy vehicles and stationary anti-aircraft batteries grow from simple points of light at very high altitude to beautifully articulated structures as the AGAV dives toward them, as its "Sapphire II" quadpulse laser cannons and "Starburst" proton missiles scream a deadly warning.

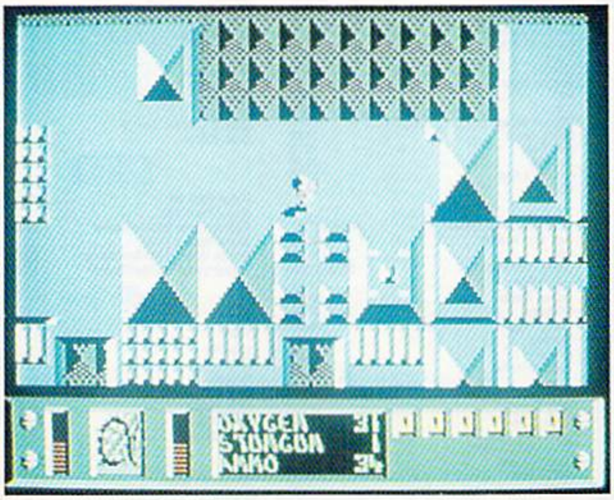
The AGAV boasts some very impressive technology. It runs on plasma energy, generated in its own onboard molecule expansion chamber, and is equipped with a "posilok" laser cell refueling nozzle on its nose. The console includes a local area scanner, energy level meter, shields, altitude meter, velocity indicator, bank level indicators, plasma drive status display, heading, and missile indicator.

But the AGAV's most fascinating gadget is its TGS (Television Guidance System). The AGAV can launch this camera for reconnaissance and continue to hover in its current position until the camera returns. It scouts the terrain and simultaneously transmits those pictures back to base.

The backdrop for the action in *Starglider* is extremely complex. The game comes with excellent documentation, including a keyboard guide, instruction booklet, poster, and a novella, without which one can hardly tell the good guys from the bad buys. The latter are in plentiful supply, too. Confronting the player are not one, not two, but three enemy alien races!

Fortunately, the program provides the gamer with the chance to get a briefing on the nature of the various enemy vehicles and weapons. When the AGAV flies into any of the docking bays scattered along the hostile landscape, the gamer may run a quick visual tutorial on enemy technology. Each tank and missile launcher telescopes into view and does a 360-degree spin while the screen displays all available data beneath.

But the main treasure of *Starglider* easily outshines all that window dressing, as good as it is. What matters here is all-out futuristic battle action, and *Starglider* delivers it, guns blazing, engines roaring, and bombs bursting. For pure visceral excitement, this is one of the year's best.



Your mission in **Parallax:** *to destroy a massive artificial intelligence which dominates an alien world, then escape to safety. You and four comrades can explore the planet either on foot or from your IBIS spacecraft.*
READER SERVICE NO. 121

At first look, *Parallax* (Mindscape) seems like yet another unremarkable *Xevious* clone: an arcade-style shoot-em-up in which the player blasts anything that moves while piloting a flying cannon. Second and third glimpses, however, reveal that there is more happening here than first appeared to be the case. *Parallax* is an original and exciting science fiction action game, with more than a dash of strategy to spice the tactical combat.

The game casts the computerist as the pilot of an IBIS spacecraft. The player and four other spacemen journey to an alien planet on what looks like an uneventful recon mission.

The destination world is divided into five zones, Alpha through Epsilon. The massive artificial intelligence which dominates this world (the C.I.U.) is planning to invade Earth. After acquiring this information, the player must make contact with the other spacefarers, destroy the C.I.U. before it springs its attack, and escape to safety through the Intergalactic Teleport in the Epsilon exit zone.

What makes *Parallax* so interesting, however, is that it permits the player to explore the alien world either in the IBIS or on foot. When the player moves to specific points in each zone, a menu appears which provides the option to leave the craft with whatever supplies (oxygen, guns, ammo, drugs, and scientists) are required.

On foot, the player can enter the Hangars, which house the scientists and computers under the protection of deadly robot guards. The player must obtain data cards from scientists and feed them into data terminals in order to decode the passwords for each of the alien zones. When the computerist deciphers the final password in the final zone, the C.I.U. unit self-destructs.

This phase of *Parallax* is balanced by some dandy combat scenes. The IBIS can actually change altitude and

engage targets on several levels.

The *Parallax* screen displays the action from a straight overhead perspective. The IBIS can whiz through the sky, land on a strip, or move through the underground out of the player's view.

The topography also includes "black holes." When the IBIS flies over one of these, it sucks the ship into its inky maw and spits it out of an adjacent "hole" with a boost in velocity. A dexterous pilot can fly into a series of these black holes to build up speed. Hyperspace Ports are similar to black holes, except that they "port" the IBIS to a totally random point.

Puzzles and challenges are sprinkled throughout this game like Easter eggs. For example, the scientists won't give up their data cards unless they've been drugged first. The only way to get past this sticky bottleneck is to carry a supply of narcotics with which to immobilize the savants.

In light of some past criticism, it is important to mention that the documentation supplied with *Parallax* is excellent. It tells computerists everything they need to get the most out of the *Parallax* experience.

Parallax also has one of the strangest sounding theme songs in computer history. It's not exactly pleasant listening, but it's guaranteed to be the first one you've ever heard quite like it.

The visuals, though nothing extraordinary, are at least workmanlike and totally acceptable. Greater realism in the animation of the IBIS ships would have improved the game. When an IBIS bounces off a zone division line, for example, it looks too cartoonish. Fortunately, the heart and soul of the gameplay, the IBIS' cannon, works very smoothly.

So don't pass by *Parallax* without a thorough inspection. It may look a little too familiar at first boot, but there's a treasure of a science fiction action game inside.

Uridium (Mindscape) is a high class throwback to a simpler age of computer entertainment. Designer/programmer Andrew Braybrook cloaks this deep space slugfest in a rudimentary story, but what *Uridium* is really about is flying, shooting, and dodging.

A fleet of super-dreadnaughts is menacing the game's home solar system. The huge interstellar craft are draining the mineral resources of the region's 15 planets for fuel. Under the player's joystick control is the last Manta fighter capable of repulsing the invaders.

The computerist begins this do-or-die mission with three lives. If the speedy and maneuverable Manta fails to eliminate a super-dreadnaught before exhausting its three lives, the game ends in an alien triumph. Otherwise, the Manta continues the battle against the next alien vessel.

Programs Covered in this Article

STARGLIDER ($39.95)
Firebird Software, P.O. Box 49, Ramsey, NJ 07446 (phone: 201-444-5700).

URIDIUM ($29.95)
PARALLAX ($29.95)
Mindscape, 3444 Dundee Road, Northbrook, IL 60062 (phone: 312-480-7667).

Before the Manta can directly attack a super-dreadnaught, it must duel the swarms of protecting fighters. These come in various shapes and sizes, and they zip back and forth across the screen at good speed.

When the Manta has thinned the protective screen of fighters sufficiently, the "land now" message appears in the upper right corner of the screen. If the Manta flies flat from left to right over the runway located at the right end of each super-dreadnaught, the Manta ship enters the fuel pod chamber.

In this subsidiary phase of the program, the display shows a pyramid of blinking lights. The player must press the joystick button at the instant the indicator with the bonus number is lit. The gamer repeats the procedure at each level of the pyramid, but must select "Quit" before the countdown clock at the top of the screen reaches zero.

The main display then returns, so that the Manta pilot can watch the once-powerful alien colossus vaporize. *Uridium* permits the Manta to make a final strafing run against the defeated foe before transporting it to the next planet and a new adversary.

The reference card which comes with the eight-page instruction manual quickly teaches the computerist to control the Manta ship with the joystick. Pushing the joystick forward or back sets the Manta's position relative to the super-dreadnaught which it is currently attacking. Pushing it left or right governs the speed and direction.

A couple of interesting special maneuvers are possible. Pushing the stick all the way in one direction produces a half-loop followed by a half-roll. This acrobatic move is especially useful for avoiding incoming missiles and mines. The Manta executes a 90-degree roll, handy for getting through tight scrapes, when the player moves the stick forward or back while pressing the action button.

The button also fires the Manta's twin cannon. Battle-hardened blast brigaders will regret the fact that each salvo requires a separate press of the button. *Uridium* is the kind of action contest which screams for continuous fire.

The graphics are good, though a little flat. The singular exception is the Manta itself, which looks great as it darts and swoops around its much larger antagonist.

Veteran computer gamers will not have to be told that *Uridium* is essentially the product of a single creative individual. Braybrook does a solid job on the play-mechanics, control scheme, and main action, but he includes few of the niceties which most software buyers take for granted. Bringing in a specialist to handle the option screens, vanity board, and other admittedly peripheral aspects of the program would have produced a more professional, "finished" program.

Real joystick jockeys won't let any lack of trimmings deter them from wading into this spaceborne shoot-out. The power of *Uridium* is its thrills, not its frills. Carnage connoisseurs, this one's for you.

Things to Come

Thanks to the timely infusion of British-designed programs, science fiction action contests are riding a new wave of popularity in the United States. Publishers are issuing two to three new futuristic disks per month.

In other words, the future couldn't look brighter for science fiction action contests. Armchair space pilots are advised to keep the rockets warm, and their trigger fingers at the ready. ☐

TIPS AHOY!

By Michael R. Davila

Send your programming and hardware hints to *Tips Ahoy!*, 45 West 34th Street – Suite 500, New York, NY 10001.

DIRECT LOADS

I never seem to be able to remember filenames. If you're like me, you frequently use the directory to load programs, by typing LOAD in front of the filename and then adding ,8: or ,8,1 at the end of the filename. Wouldn't it be nice if I could just type LOAD in front of the filename, press RETURN and have ,8: or ,8,1 already there?

I knew it was possible to embed characters to the right of the filename by separating the filename from the embedded characters with a SHIFTed space (when embedded, COMMODORE-D turns out to be a comma, and the shifted @ a colon). I could of course rename each file one by one and remember the proper sequence to add ,8: or ,8,1, but that seemed like too much trouble.

I modified a program that appeared originally in *Tips Ahoy!* (*Ahoy!,* February 1986, page 87). *Directory Highlighter* was designed to highlight a filename in the directory. A few modifications allowed me to use this program to add the ,8: or ,8,1 suffix. The addition of this suffix still allows 12 characters for my BASIC program filenames and 11 characters for my machine language program filenames.

–Benoit Beaudoin
Montreal, Canada

```
•10 OPEN15,8,15:GOSUB160
•20 SS$=CHR$(160):CD$=CHR$(172):N8$=CHR$(
56):SA$=CHR$(186):N1$=CHR$(49)
•30 BA$=SS$+CD$+N8$+SA$
•40 ML$=SS$+CD$+N8$+CD$+N1$
•50 PRINT"[CLEAR][RVSON]B[RVSOFF]ASIC OR
[RVSON]M[RVSOFF]ACHINE LANGUAGE
•60 GET KY$: IF KY$="" THEN 60
•70 IF KY$="B" THEN 100
•80 IF KY$="M" THEN 130
•90 END
•100 INPUT"NAME OR BASIC PROGRAM";BP$
•110 PRINT#15,"R0:"+BP$+BA$+"="+BP$:GOSUB
160
•120 GOTO50
•130 INPUT"NAME OF ML PROGRAM";MP$
•140 PRINT#15,"R0:"+MP$+ML$+"="+MP$:GOSUB
160
•150 GOTO50
•160 INPUT#15,E,E$,T,S
•170 IF E=0 THEN RETURN
•180 PRINT E,E$
•190 END
```

SINE DRAWING ON THE C-128

The SIN mathematic function is a very old yet powerful tool in BASIC. With the power and speed of the 128's BASIC 7.0, the SIN can be used to create complex designs.

The program below uses three main variables that you must answer at the INPUT prompt. They are for the X and Y axes, and the degree at which the picture is to be calculated and drawn. The program makes use of the 128's DRAW command also. The DRAW command uses a different kind of technique, as you will see in lines 50 and 64. DRAWTO? I know, the 128's documentation does not make mention of it; however it is the equivalent of DRAW 1, RDOT (0), RDOT (1) TO X,Y, where RDOT(0) and (1) are the last X and Y coordinates drawn at. Thus DRAWTO simply means to DRAW a line from the last X and Y location to an updated position.

When you run the program, experiment with different X and Y axes and the viewing angle. There are many, many possible screen formations!

–Robert J. Tiess
Middletown, NY

```
•14 COLOR 0,1 : COLOR 4,1 :COLOR 5,12
•16 GRAPHIC 0,1 : C=0 : Z=0 : B=0 : U=0
•18 INPUT "HORIZONTAL AXIS";H
•20 INPUT "VERTICAL AXIS";V
•22 INPUT "VIEWING ANGLE 0 TO 360";P
•24 RA=57.2957
•26 NT=90
•28 NF=95
•30 OT=120
•32 TS=127
•34 E=8
•36 FOR I=1 TO 90 : NEXT
•38 P=P / RA
•40 GRAPHIC 1,1
•42 IF V >= H THEN M=V
•44 IF H > V THEN M=H
•46 Z=Z+H * (E/M)
•48 B=B+V * (E/M)
•50 X=INT((SIN(Z/RA+P)*OT)+TS)
•52 Y=INT((SIN(B/RA)*NT)+NF):IF U=0 THEN
LOCATE X,Y : U=1
•54 C=C+1
•56 IF C=< 4 THEN 64
•58 DRAWTO X,Y
•60 GETA$ : IF A$=" "THEN 68
•62 GOTO 46
•64 DRAWTO X,Y
•66 GOTO 46
•68 GRAPHIC 0,1
•70 GOTO 16
```

COLORFUL 64

If you think the Commodore's 16 colors aren't enough, you may like this. This short program will allow you to use 29 different background colors. It creates a custom character in the shape of a checkerboard. When this character is printed, it will blend with the background to make a new

EMERALD COMPONENTS INTERNATIONAL

DISK DRIVES

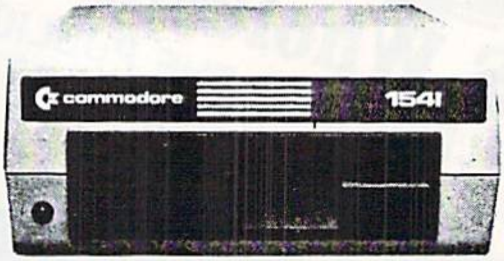
FSD-2
EXCELERATOR PLUS
Commodore Compatible
DISK DRIVE



Direct Drive!
Faster, Quieter
and More Reliable.
Full One-Year Warranty.

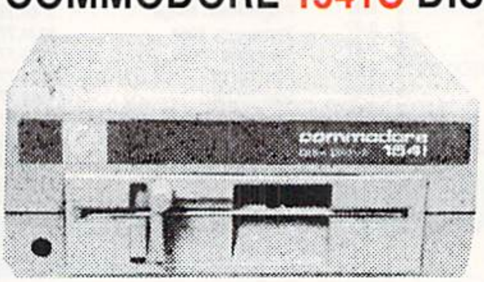
$159

COMMODORE 1541 DISK DRIVE



$142

COMMODORE 1541C DISK DRIVE



$169

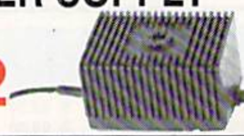
HARDWARE & ACCESSORIES

COMMODORE 64 COMPUTER



$129 With specially priced software package

COMMODORE POWER SUPPLY



$32

MODEM
300 Baud



$28 SAVE OVER $70!

MOUSE



$36

HEAD CLEANING KIT



$11.95

JOYSTICK



$9.95

ALL MAJOR CREDIT CARDS ACCEPTED

VISA · AMERICAN EXPRESS · MasterCard · DISCOVER

6 AM TO 6 PM
6 DAYS A WEEK

Emerald Components International
P.O. Box 1441 / Eugene, OR 97440
In Ore., Call: 503-683-1154
Customer Service: 503-683-1154
For further information, contact your local Commodore dealer.

ECI 1-800-356-5178

shade. If you print it in red, you will have a dark red, because the checkerboard will let some of the background color come through.

When you run the program, all this will seem a lot clearer. The program will print out 29 different colors, including 8 shades of gray, 6 shades of red, and 4 shades of green. If you fool around with different background and text colors, you can create some interesting combinations.

When you first run the program, you will have to wait about a minute while it transfers the character data.

—Chris Jones
Oak Ridge, TN

```
·5 PRINT"[CLEAR][HOME]LOADING DATA[3"."]"
 : POKE52,48:POKE56,48:CLR
·10 POKE56334,PEEK(56334)AND254:POKE1,PEE
K(1)AND251:FORI=0 TO 2040
·15 POKE(12288+I),PEEK(53248+I):NEXT:POKE
1,PEEK(1)OR4:POKE56334,PEEK(56334)OR1
·20 POKE53272,(PEEK(53272)AND240)+12
·25 FORG=0TO7:READ A:POKE 13080+G,A:NEXT
·30 POKE53281,0:POKE53280,0
·35 PRINT"[CLEAR][WHITE]29 POSSIBLE COLOR
S"
·40 PRINT"[5" "][WHITE][RVSON][5" "][RVSO
FF][RED][5"[c T]"][BLUE][5"[c T]"][PURPL
E][5"[c T]"][GREEN][5"[c T]"][YELLOW][5"
[c T]"]"
·45 PRINT"[c 4][5"[c T]"][5" "][c 3][5"[c
 T]"][c 7][5"[c T]"][PURPLE][RVSON][5" "
][RVSOFF][c 6][5"[c T]"][YELLOW][RVSON][
5" "][RVSOFF]"
·50 PRINT"[c 5][5"[c T]"][5" "][c 1][5"[c
 T]"][CYAN][5"[c T]"][5" "][GREEN][RVSON
][5" "][RVSOFF]"
·55 PRINT"[c 8][5"[c T]"][5" "][RED][RVSO
N][5" "][BLUE][5" "][RVSOFF][5" "][c 6][
RVSON][5" "]"
·60 PRINT"[WHITE][5"[c T]"][5" "][c 3][RV
SON][5" "][c 7][5" "][RVSOFF][5" "]"
·65 PRINT"[c 4][RVSON][5" "][RVSOFF][5" "
][c 1][RVSON][5" "][CYAN][5" "]"
·70 PRINT"[c 5][RVSON][5" "][RVSOFF][5" "
][c 2][RVSON][5" "][RVSOFF]"
·75 PRINT"[c 8][RVSON][5" "][RVSOFF]"
·80 PRINT"[HOME][11"[DOWN]"]"
·85 PRINT"[WHITE]8 GRAYS  7 REDS  6 BLUES
  2 PURPLES"
·90 PRINT"4 GREENS  2 YELLOWS"
·95 DATA 85,170,85,170,85,170,85,170
```

CHASER MARQUEE

Would you believe a 12-line program that creates a marquee of chasing lights around the border of the screen?

You can make your own in assembly language, or you can tack this nifty little routine I created onto the beginning of any BASIC program to add a touch of class.

The routine uses extended background color mode, so you are limited to the first 64 characters for your titling.

The FOR-NEXT loop in line 130 is a delay loop; changing the value of "X" will control the speed at which the lights run around the perimeter. You can also POKE values other than red and white into the color registers in line 90 for different effects. Try gray and white or two blacks and a bright color for neat variations. There are three registers, and the colors you POKE into them will chase after one another around the edge until you push any key to continue with the rest of the program.

A must-have for onscreen board games or displays!

—Cleve Blakemore
Richmond, VA

```
·10 PRINTCHR$(147):V=53248:POKEV+32,0:POK
EV+33,0
·20 FORX=0TO37STEP3:POKE1024+X,64+32:POKE
1024+X+1,128+32:POKE1024+X+2,192+32
·30 POKE1984+X,192+32:POKE1984+X+1,128+32
:POKE1984+X+2,64+32
·40 NEXT
·50 FORX=0TO22STEP3
·60 POKE1024+X*40,192+32:POKE1024+(X+1)*4
0,128+32:POKE1024+(X+2)*40,64+32
·70 POKE1063+X*40,64+32:POKE1063+(X+1)*40
,128+32:POKE1063+(X+2)*40,192+32
·80 NEXT:POKE2023,128+32
·90 POKEV+17,PEEK(V+17)OR64:POKEV+34,1:PO
KEV+35,2:POKEV+36,2
·100 POKE214,7:PRINT:PRINTTAB(12)CHR$(31)
"YOUR TITLE HERE"
·110 A=PEEK(V+34):B=PEEK(V+35):C=PEEK(V+3
6)
·120 POKEV+34,C:POKEV+35,A:POKEV+36,B
·130 GETA$:IFA$=""THENFORX=1TO75:NEXT:GOT
O110
```

NUMBER ROUNDING

While writing antenna designer programs I became frustrated with trying to insert numbers into lines of text. Almost invariably the numbers would have 5 or more decimal places, making it impossible to maintain neat lines. Since such accuracy is neither necessary nor possible to maintain when cutting the antenna, I began to look for a way to round those numbers to one or two decimal places. The integer function provides an easy way to round a number down to the next lowest whole number. The expression A=INT(A+.5) will round to the nearest whole number. Rounding to one decimal place then becomes A=INT(A*10+.5)/10. This formula can be extended to round to as many decimal places as desired. The short program below will calculate the correct length of a half wave dipole antenna and round that length to one, two, and three decimal places.

—Henry F. Smith
APO Miami, FL

```
·10 REM NUMBER ROUNDER
·20 PRINT"[CLEAR][DOWN][DOWN][8" "][RVSON
]HALF WAVE DIPOLE DESIGNER[RVSOFF]"
·30 INPUT"[DOWN]  FREQUENCY  IN MHZ ";FR
·40 A=468/FR
```

ATTENTION C-128 OWNERS!

FEEL NEGLECTED? NO 80-COLUMN GRAPHICS SOFTWARE?

The Commodore C-128 is the most powerful and flexible 8-bit microcomputer yet released to the public. Unfortunately, there is little software available for its native 80-Column mode; and most of that is Productivity software using only the Text Character Display, not the true 80-Column Hi-Res Graphic Display which the machine is capable of producing.

Commodore originally intended the C-128's 8563 Video Display Chip to support 80-Columns only in Text mode, not Graphics. While standard C-128 Basic takes full advantage of the 40-Column graphics capability of the machine, there is almost nothing which allows the Basic programmer access to the 80-Column Graphics mode (Yes, there is an 80-Column Graphics mode!).

TIRED OF WAITING FOR YOUR 128 TO COME ALIVE? WAIT NO MORE. BASIC 8 IS HERE!

Patech Software is proud to introduce **Basic 8** with **Basic Paint**, the first C-128 software package specifically designed to unleash the hidden graphics potential of your Commodore C-128. Using a special wedge technique, **Basic 8** achieves performance rivaling that of 16-bit micros! Imagine your 128 producing resolution of 640 x 200 in monochrome and 640 x 192 in 16 colors without additional hardware! **Basic 8** provides the Basic programmer with the most powerful and productive graphics system ever developed for an 8-bit microcomputer!

A NEW HI-RES 3-D GRAPHICS LANGUAGE FOR THE C-128

Basic 8 adds over 50 Hi-Res graphics commands to Basic 7.0 and is completely compatible with its advanced non-graphic commands. You work in a true 3-Dimensional environment, controlling such parameters as perspective, viewing angles and the origin & depth of the view. Just select one of many graphics modes and draw 3-D lines, dots, circles, boxes, variable arcs and a multitude of complex solid shapes with a single command.

Switch easily between any of the 32 pre-defined graphic screens, scroll a screen, save and load multiple screens as brushes, define a logo, paint with exotic color patterns of your own design and print hard copy of your screens in various sizes and in color! We've even added commands for graphic cut & paste functions, brushes, windows and fonts along with ultra-sophisticated concepts like multidraw, grow, shearing, scaling and much, much, more.

More than a mere extension of Basic 7.0 – **Basic 8** is a total Development System which provides a whole new and exciting programming environment. Chances are that if your graphic imagination can see it, **Basic 8** will let you do it!.

UNHEARD-OF RESOLUTION AND STUNNING COLOR DISPLAYS ON YOUR C-128!

Pixel Resolution (screen size) and Color Resolution (color cell size) are determined by available Video RAM. The C-128 has 16K. This can be upgraded to the 64K of the C-128D.

The unexpanded C-128 produces a full 640 x 200 screen in monochrome and a maximum of 640 x 192 in color with an 8 x 16 dot cell. Increasing color resolution (smaller cell) decreases screen height. (Dot size remains the same throughout.) The most useful screen is 640 x 176 with an 8 x 8 cell. That's still pretty impressive! (The C-64 has 320 x 200 with the same cell.) If you use a 640 x 152, you can double the color resolution to 8 x 4. The resulting displays are absolutely stunning!

The 64K Video RAM allows the full 640 x 200 screen with an 8 x 2 cell (we doubled it again, this time with a full screen!) Several such screens can reside in Video RAM simultaneously, each with different resolutions. You can view one screen while working on another and create Virtual Screens (larger than the displayable 640 x 200) in Video RAM.

BASIC PAINT – AN 80-COLUMN COLOR DRAWING PROGRAM!

To demonstrate the power and versatility of this new graphics language, we have created **Basic Paint**, a flexible icon-based, mouse driven, 80-Column color drawing application with menus, fonts, brushes, patterns and requestors. **Basic Paint** is written in **Basic 8**, so that screens you create with it may be saved for use in your programs.

BASIC 8 GROWS WITH YOUR SYSTEM

Basic 8 supports all C-128 hardware upgrades and expanders. Each time you up-grade your system, **Basic 8** is ready and waiting with even more graphic power than ever!

Special printing features include color printing, 90° rotatable screens and variable dot density selection. The 64K Video RAM adds printing of entire Virtual Screens with the full page being dot addressable. Results are limited only by the capabilities of your printer.

★★ ALL THIS GRAPHICS POTENTIAL ★★ IS YOURS AT THE INTRODUCTORY PRICE OF $39.95

Includes: A 180-page Users Manual, the unprotected **Basic 8** Program Disk which creates **Basic 8** Work Disks (with complete Editor System), **Basic Paint** Work Disks and **Basic 8** RunTime Disks (lets you load and run your creations independent of **Basic 8** using the included WORKBENCH utility).

Requires: C-128 or 128D, 80-Column RGB Monitor & at least one Commodore compatible Disk Drive.

Supports: Expanded Video RAM (to 64K), Both RAM Expanders, The 1351 Proportional Mouse, Joysticks and most Dot Matrix Printers

BECOME A PUBLISHED SOFTWARE DEVELOPER! ENTER THE BASIC 8 PROGRAMMING CONTEST. WIN THE GRAND PRIZE OF $1,000.00 AND PATECH SOFTWARE WILL PUBLISH YOUR PROGRAM! Three Other Cash Prizes To Be Awarded. Details Included In The Basic 8 Package.

PATECH Software, Inc.

Mail your order to:
Computer Mart, Dept. S • 2700 NE Andresen Road • Vancouver, WA 98661
Phone orders welcome: 206-695-1393
Same day shipping/No C.O.D. orders outside U.S.
CHECKS, MONEY ORDERS OR VISA/MASTERCARD.
PLEASE NOTE: Free shipping & handling on all orders • C.O.D. add $3.00 to total order • All orders must be paid in U.S. funds.

VISA MasterCard

```
•50 REM ROUND TO TENTHS
•60 B=INT(A*10+.5)/10
•70 REM ROUND TO HUNDREDTHS
•80 C=INT(A*100+.5)/100
•90 REM ROUND TO THOUSANDTHS
•100 D=INT(A*1000+.5)/1000
•110 PRINT" DIPOLE LENGTH IS[7" "]"A" FEE
T"
•120 PRINT" ROUNDED TO TENTHS[6" "]"B"[3"
 "]FEET"
•130 PRINT" ROUNDED TO HENDREDTHS  "C"  F
EET"
•140 PRINT" ROUNDED TO THOUSANDTHS "D" FE
ET"
```

RESET TO 64 MODE

The reset switch on the C-128 is a very nice feature, especially in the C-64 mode where you must perform a reset to gain control of the machine after using many commercially available software titles. Unfortunately, however, in order to go back to the C-64 mode, you must also hold down the COMMODORE key while pressing the reset switch. Since my family only uses the C-64 side of the computer, they felt it was an inconvenience to press both the COMMODORE key and the reset switch. So I wrote the program below, saved it on disk as "RESET TO 64 MODE", and told them to enter (from the C-128 mode): RUN"RESET TO 64 MODE".

When the program is run, it will GOTO the 64 mode. But more important, only the reset switch need be pressed to return to the 64 mode. —Shawn K. Smith
Bronx, NY

```
•0 REM RESET TO C64   BY  SHAWN K. SMITH
•1 BANK1:POKE65528,77:POKE65529,255:GO64
```

FREE 'N EASY

It's definitely handy to be able to check the bytes free on your 64, especially when using memory-eating monsters like arrays. As you know, PRINTFRE(0) is built into the machine for keeping track of memory usage. Unfortunately it has two drawbacks. First of all, if you have more than 32767 bytes free, the 64 merrily displays a negative number. When this happens, you'll have to type PRINTFRE(0)−65536*(FRE(0)<0) in order to get an accurate reading. Secondly, if you check the bytes free with relentless regularity, your poor fingers will be worn to a frazzle in no time. There is a better way. *Free 'n Easy* quickly and accurately displays the bytes free, no matter what their number.

After saving a copy of *Free 'n Easy*, run it. This POKEs the data into memory. To switch it on, type SYS700, then press RETURN. You switch it off the same way. The first SYS turns it on, the second turns it off, etc. Now you can load, save, and work on whatever you want.

Continued on page 98

Featured This Month:

Championship Baseball
GFL Championship Football......53

Marauder II......................54

Amiga Trackball..................57

CHAMPIONSHIP BASEBALL
GFL CHAMPIONSHIP FOOTBALL

Gamestar
Amiga
Disk; Baseball $39.95, Football $44.95

Although other publishers produce action-strategy sports games, Gamestar has set the standard against which all other similar programs must be measured. The company has now issued two of its best team sports titles for the Amiga.

Although many notable designers and programmers have worked on Gamestar's titles, including the pair discussed here, much credit goes to the company's president and chief designer, Scott Orr. His vision of what an action-strategy sports simulation should be transcends the details of any individual Gamestar disk program.

As a group, *Championship Baseball* and *GFL Championship Football* are easy to learn, yet require practice to master. Both rely primarily on joystick or mouse order entry and employ multiple screens to catch all the action and the strategy which underlies it. They both have superb animated graphics, thorough documentation, and tremendous replayability.

These "family traits" guarantee that Gamestar creations will be good, solid entertainment. But it is their wealth of detail, special features, and ability to capture the essence of the sports which they simulate that make them great computer games.

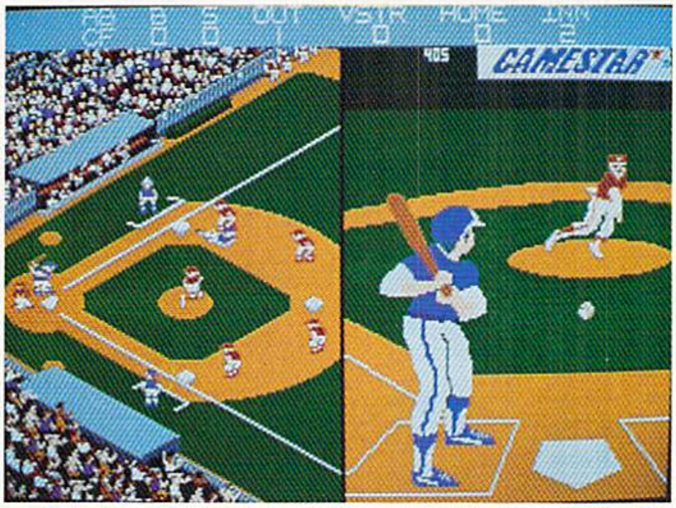
Championship Baseball brings new sophistication to arcade-style hardball. Although the user can play an exhibition with a prefabricated team or take batting practice, options allow the armchair manager to field customized squads in a league play framework.

Drafting a team is surprisingly painless. Menus give alternatives for each regular and pitcher, plus a small selection of utility men. The hitting style (liner or slugger) is shown next to each fielder, along with his ratings for batting, catching, running, and throwing. Pitchers have three ratings: speed, control, and stamina. As a special kick, the computerist can name each team member. There's nothing like steamrolling one of *Championship Baseball's* four divisions with a team of old girlfriends or Irangate conspirators.

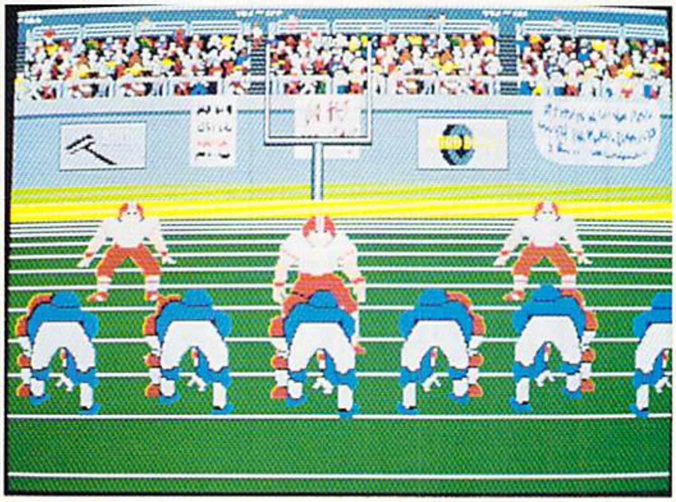
All this is only preamble to an outstanding action contest. A split-screen display shows an overhead view of the field on the left, and a close-up of the batter-pitcher confrontation from the catcher's viewpoint on the right. If the batter hits the ball, the field display fills the entire screen to facilitate catching the ball, throwing it to the right base, and controlling the runners.

Holding the button and pushing the stick in one of eight directions orders one of eight pitching deliveries. The defensive manager in this one- or two-player game can toss fastballs, curves, sliders, screwballs, and knucklers. The offensive manager times the pitch and, using the joystick, swings away or drops a bunt.

While the ratings definitely color the action, hand-eye coordination and reflexes dominate play. The joystick command control scheme lets the onscreen players take leads, steal, pick off runners, slide into bases, and much more. *Championship Baseball* feels real and moves fast, two outstanding qualifica-



Championship Baseball's *split screen offers overhead and catcher's-eye views, changing to a full field perspective when the batter hits the ball. READER SERVICE NO. 164*



GFL Championship Football's *secondary screens provide overall field view, but first person perspective play execution is the most notable. READER SERVICE NO. 165*

tions for an action baseball simulation.

GFL Championship Football can also simulate a season of league action, but its most notable innovation is the first-person perspective. The secondary screens provide the overall field perspective needed to follow the flow of the game, but the real excitement is the dynamically animated play execution screen.

Before a team can run a play from scrimmage, the coach must call an offensive or defensive play for the next down. A pair of menus, one each for offense and defense, list the possibilities. Pushing the joystick highlights each choice in turn, and pushing the button locks in the command.

That's when *GFL Championship Football* moves into overdrive. The graphics put the computerist right on the field as the running back or receiver on offense. As a running back, for example, you take the handoff, scoot laterally behind your line of trusty computer-controlled blockers, and hit a hole or swing around end. You can snap the joystick handle to stiff-arm a would-be tackler, but unless your man scores a touchdown, a burly defender's lunge is the last thing you'll see before the whistle signals the end of the play.

GFL Championship Football sounds even better than it looks, if that's possible. Each play from scrimmage is a symphony of thundering footsteps, gargantuan grunts, and the teeth-rattling crunch of bodies colliding.

Don't expect to breeze to the league title the first time. *GFL Championship Football* is probably the most demanding action sports simulation ever produced. Yet if any computer game is worth a couple of hours of study, this is it. *GFL Championship Football* is a Super Bowl champion among action football titles.

Activision, 2350 Bayshore Parkway, Mountain View, CA 94043 (phone: 415-960-0410).

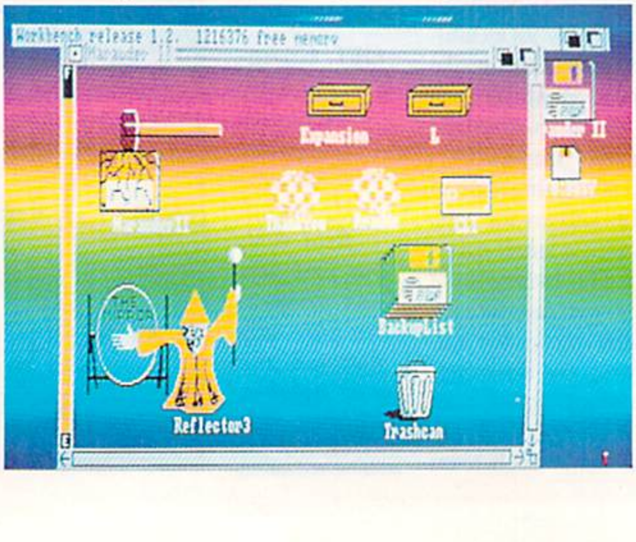
—Arnie Katz

MARAUDER II
Discovery Software International
Amiga
Price: $39.95

Some things never change. No sooner does a new computer appear on the market than it is followed by third-party copy protected software. This software is immediately followed by a series of copy programs designed to duplicate the aforementioned copy protected software and to give the users back their government-granted right to create archival copies of their owned software.

More often than not, in what seems to be an exercise in contradiction, the copy programs themselves are copy protected. This of course prompts the producers of subsequent copy programs to show their prowess by providing the capabilities to copy the copy protected copy programs. And so on and so on.

The Amiga has not escaped this syndrome. From the day of its introduction we have found copy protected software to be readily available. These programs were closely followed by *Marauder* from Discovery Software International. In spite of its rather rambunctious name, *Marauder* did a reasonable job of copying the software available at the time. However, as the copy protection on the Amiga matured, it was gradually left behind.



Yes, that's a Workbench screen. To get it, boot up **Marauder II**, *switch screens with the left-Amiga and n or m combo and pull down the Workbench screen a hair.* READER SERVICE NO. 166

PHOTO: MORTON KEVELSON

The people at Discovery Software did not remain idle. Following a brief hiatus they presented *Marauder II* (seems everybody is into sequels these days), probably the most effective copy program for the Amiga available at this time.

Marauder II has all the earmarks of a successful copy program. It has a flashy screen to which the accompanying screen shot does not do justice. The rainbow of colors continuously scroll up the display. It generates a startling digitized sound when it boots up. It has a separate routine for copying the copy protected disk of its closest competitor, thereby clearly demonstrating its superiority. It is properly adapted for the Amiga with an easy to use, icon-based, mouse-driven interface, and it supports multitasking. Most important of all it is easily upgradable at reasonable cost, which permits it to indefinitely forestall its own obsolescence.

The key to *Marauder II's* upgradability is also the basis of its intelligence. Rather than attempting to copy via brute force duplication, *Marauder II* utilizes a "brain module" for the basis of its operations. The brain module represents the intelligence of its programmers. It automatically recognizes a host of copy protected programs and provides the data which allows *Marauder II* to duplicate this program, and in most cases also strip the copy protection at the same time. In essence, *Marauder II* is a parameter copier with a built-in parameter list.

The brain module is a standard AmigaDOS file which may be easily installed by the user. New brain modules are available to registered owners for a $15 fee. This fee covers the cost of developing new parameters, which is done on a continuous basis. Brain modules can be ordered by mail, or they can be downloaded from Discovery Software's own bulletin board. It is not necessary to purchase every issue of the brain module to be fully up to date. Each brain module incorporates all the program parameters of all the preceding brain modules.

Marauder II also offers several mundane advantages over the AmigaDOS Diskcopy command. To begin with, *Marauder II* can make an exact duplicate of a disk, something that Diskcopy does not do. *Marauder II* is also slightly faster than Diskcopy, at 87 seconds versus 102 seconds more or less. *Marauder II* is also capable of making up to four simultaneous copies of a disk. Of course, your Amiga will need three external disk drives as well as the internal drive, and at least one megabyte of expansion RAM. If your system has the extra RAM, then the disk can be read into memory and multiple copies can be generated forever after without any further disk reads. Even without the extra memory, you can produce up to three simultaneous copies if you have the external disk drives.

Although *Marauder II* is fully automatic and works best with its default settings, there are several options available on its pull-down menus. The default analytical setting may be defeated by choosing the verbatim option. This turns off the program's ability to analyze the original disk and generates an exact duplicate if possible. You may also turn on the verify option, which checks the destination copy for deviations from the original. This may be useful for detecting a damaged destination disk. Some esoteric operations may also make use of *Marauder II's* ability to synchronize with the disk's index hole while copying.

Marauder II includes several standalone utilities on its Workbench disk, not all of which are described in the manual. The *Marauder II* manual also forms the basis of its own copy protection. When you first boot up you will be required to enter a word from a specific location in the manual. The specific word changes each time you boot up. Getting back to the standalone utilities, if you examine both the root directories and the c directory you will find the following:

c/IDent: File Identifier. Reports the file name, file size, checksum and date information.

c/Peek: Displays the hex and ASCII contents of a file. Peek is similar to the AmigaDOS Type command, with hex option enabled, except it lets you specify the starting position in the file and the number of bytes to display.

c/XFind: Locates a specified pattern of hex bytes and reports on the location of each match.

Diskwipe: Completely and quickly clears an entire disk. This is slightly faster than reformatting the disk.

Decoder: Completely decrypts early Electronic Arts software to permit faster loading.

DiskErr: Scans an entire disk and reports all errors.

Most of these utilities were put on the *Marauder II* distribution disk to permit the decrypting and copying of new software by remote control. If you should come across a program which *Marauder II* cannot copy and if you happen to give Discovery Software a call, you may be instructed on the use of these utilities on the recalcitrant software. If the conditions are right you will be rewarded with a working copy and Discovery Software will have a new parameter for the brain module.

Multitasking is possible with *Marauder II*. The underlying Workbench screen may be accessed with the left-Amiga and n or m key combinations. Of course disk operations are not possible, as *Marauder II* takes control of the disk drives. *Marauder II* may be easily installed on any Workbench disk using the provided utility.

Marauder II is a worthwhile investment for serious Amiga users. As of this date, Discovery Software has already issued seven brain module updates. Updates have been coming at the rate of once a month. These people are certainly dedicated to their task.

Discovery Software International, 903 East Willow Grove Avenue, Wyndmoor, PA 19118 (phone: 215-546-1533). *—Morton Kevelson*

AMIGA TRACKBALL

Zebra Systems, Inc.
Amiga
Price: $49.95

If you want to really make mousetracks with your Amiga, then replacing its mouse with a trackball may be just the way to go. We have in general been most satisified with the mouse as an input device for the Amiga. We found it takes very little practice to become adept at its tail twitching and ear clicking. However, we have been encountering some difficulty with the other denizens which inhabit the desktop upon which our Amiga resides. These creatures have developed the insidious habit of encroaching upon the territory which our mouse has reserved for its own use.

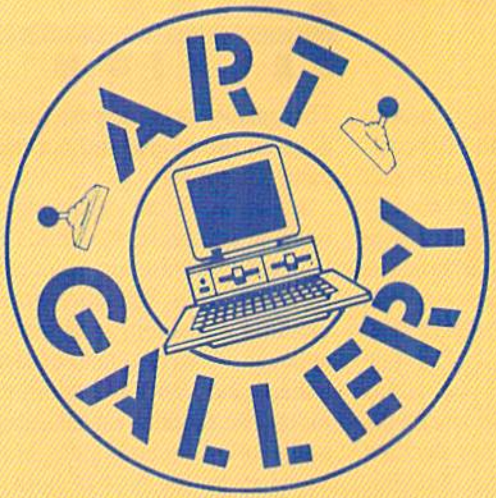
We have found the Amiga's mouse to be an extremely territorial beast. When crowded it will frequently react by leaping off the desktop and thereby becoming nearly useless. Zebra Systems has come up with an alternative to the territorial mouse. It seems that WICO, the well-known maker of sturdy joysticks and trackballs, has left that field of endeavor. Zebra Systems has acquired a quantity of the trackballs and in a fit of ingenuity has decided to modify them for use with the Amiga. Since the hardware of a trackball is essentially an inverted mouse, it was only necessary to design a single circuit board to replace the one made by WICO.

The requirement for two push buttons means that the modified trackballs were originally intended for the Apple computers. On this model a single large push button is situated at the upper left hand corner of the trackball. This button has been wired up to correspond to the left mouse button. A second, smaller button is situated just below the first on the left side. This button corresponds to the right mouse button. In addition, Zebra Systems has installed a jack and adaptor cable to allow the mouse to be plugged into the trackball. This enables only the push button functions on the mouse and not the movement functions.

In use we found the arrangement to be most agreeable. As a rule the trackball worked best for point and click type operations. The click-drag-release function was slightly awkward at first, although we were able to master it with either hand. With the addition of the mouse we found the availability of the push button on both sides of the keyboard to be the best of both worlds. Elimination of the roaming mouse solved some of our desktop territorial problems. We also noticed that both the mouse and trackball cables were long enough to reach around the back of the computer, eliminating some of the clutter around our keyboard.

The supply of surplus modified WICO trackballs is definitely limited, so you may have to hustle to get one. Alternatively, if you already own a trackball and you have some ability with electronic construction, we'll supply a schematic next month which may be sufficient to let you modify your own.

Zebra Systems, Inc., 78-06 Jamaica Ave., Woodhaven, NY 11421 (phone: 718-296-2385). *—Morton Kevelson*



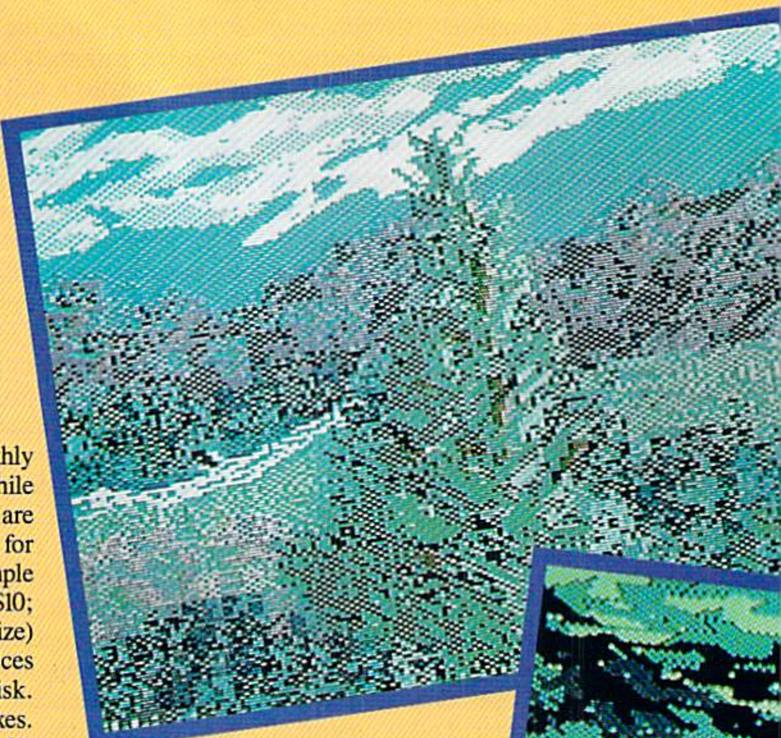
Art Gallery Disk Sale

The images on these pages are now available on a monthly disk. Multicolor images are supplied in *Koala* format, while high-resolution images are in *DOODLE!* format. Included are a slide show for easy viewing, along with a bit map dump for your 1525 printer or properly interfaced equivalent. A sample *Art Gallery* disk with slide show and printer dumps is $10; or send a stamped and self-addressed envelope (business size) for a listing of available *Art Gallery* collection disks. Prices shown are for US and Canada. All others add $3 per disk. New York State residents please add appropriate sales taxes. Disks may be ordered from Morton Kevelson, P.O. Box 260, Homecrest Station, Brooklyn, NY 11229.

Contribute to Ahoy!'s Art Gallery

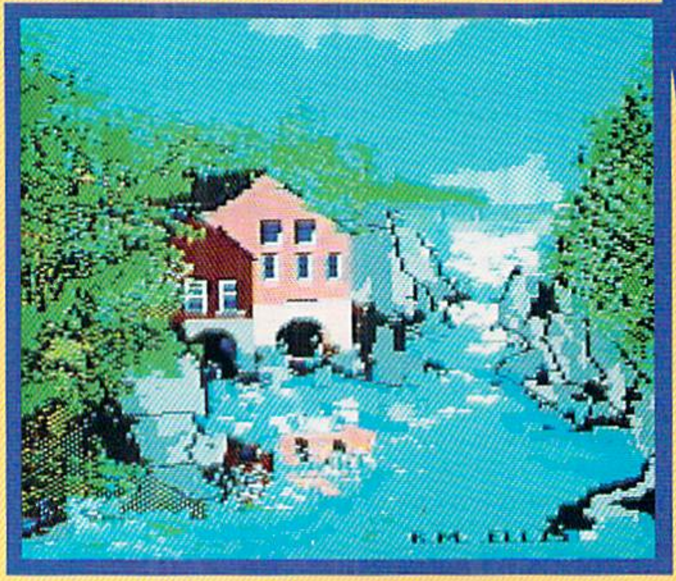
The *Ahoy! Art Gallery* offers the opportunity for fame and fortune to any and all aspiring Commodore artists. Simply send Morton (see address above) your work on disk indicating the drawing package or file format of the images. Inclusion of a self-addressed post card will guarantee an immediate response. All graphics produced on the C-64/ C-128, Plus/4, and Amiga computers are eligible. In exchange your work will receive the opportunity for display in these pages. All published works will receive royalties based on the monthly *Art Gallery* disk sales. Also, both published and unpublished images may be included on *Art Gallery* collection disks.

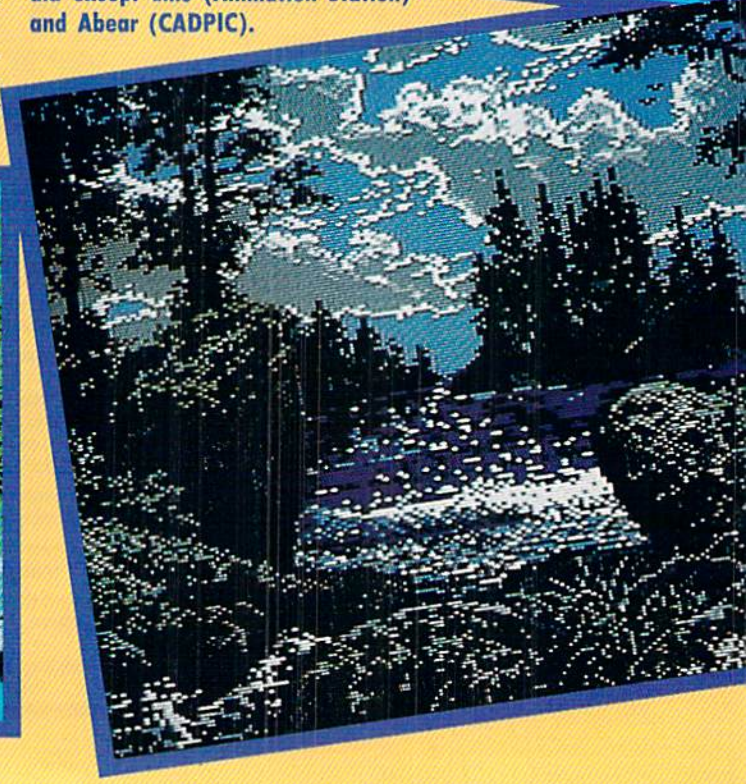
Note that the *Art Gallery* is not a contest. Published pictures are selected in an arbitrary and capricious fashion by the *Ahoy!* Art Director based solely on the artistic merit of the individual images.





In case you can't move your computer to the veranda for these summer months, we've brought the great outdoors indoors. Top row: *Fall* by Nikki Lewis (Des Plaines, IL); *Ballagio* by Albert Valsecchi (Milano, Italy). Middle: *River* by Barri Olson (Madison, WI). Bottom: *Place Jacques-Cartier* by Robert M. Ellis (Quebec, Canada); *Island* by Gerald J. Abear (Las Cruces, NM); and *Farmer* by A. Black (Santa Rosa, CA). All artists used Koala except Ellis (Animation Station) and Abear (CADPIC).













If I have any strong memories from my youth, they are probably of the arcades. There always seemed to be an incredible, surrealistic adventure waiting within the confines of the darkened, carpeted interiors: a great struggle between good and evil I could not find in the real world, always only a quarter away.

I was never much interested in the standbys like *Defender* or *Pac-Man*, but confess to having been in love with the old classic *Venture*. You remember–the little smiley face that battled a host of vile fiends with only hits wits and arrows, descending deeper and deeper on a lone journey into the netherworld in quest of glory below. What courage! What cunning! I saw volumes of character carved in the lines of that simple, confident grin.

They say the best games are the ones you make for yourself, and I admit the game *Archer* was inspired by that swell arcade hit, now long since gone to its resting place in the silicon graveyard.

This is no bare-bones game, either. It utilizes 25 sprite shapes, 3 interrupt driven songs, fast machine language aided action, 10 dungeon levels and 5 different floor plans, each with assorted creatures, traps, and puzzles. It also has an intriguing title screen with a smooth scrolling backdrop.

The program is a hybrid BASIC 7.0/machine language game, and the movement is smooth and exciting. You'll need a high quality joystick with good movement in the diagonals plugged into Port 2 to play.

The program will take a moment to load the interrupt and sprite data before the title screen appears. Press the fire button to start the game.

The first thing you'll always see is a large overview of the dungeon level you're on first, with the little green face representing you. Guide the Archer to the doors of each of the rooms in the dungeon while dodging Loog and Moog, two troll sentries you'd do well to avoid, because your arrows will never pierce their rough hide.

Upon your first entry into any dungeon level, the floor will be littered with a sampling of diamonds which you can pick up as you go for bonus points, each diamond worth LEVEL×50.

When you enter the door of the rooms, the screen will blank for a moment, then cut to a blown-up interior of the inside of that room, and a blown-up version of the Archer, complete with bow and arrow. To move, simply push the joystick in the desired direction. To fire, point and push the button.

Each dungeon has several different rooms, each possessing a different treasure. You must penetrate each room and collect the treasures before the doors to the staircase at the upper left will open, enabling you to descend to the next level. Each time you pick up a treasure, it will appear in the "booty" bracket at the bottom of the screen. Notice that when you reemerge into the large scale map, Loog and Moog have gathered up any diamonds remaining.

The value of the treasures you collect depends on the level and type of menace in the room. Each dungeon creature you kill is worth LEVEL×1000. You can elect just to collect treasure, or go for the points and try to clean out the rooms of monsters as you progress.

There are more than monsters to fight here. There are also "alcoves," little traps with treasure inside guarded by shifting walls, or rotating panels. Timing and precision are required to survive them.

If you make it to the tenth level, you will definitely be one of the elite, because I have only made it once myself as of this writing. Things get really frantic in the more advanced levels of the dungeon, requiring anticipation and accurate, cobra-fast reflexes.

You start the game with three men, with a possible bonus man when you make it to level five. The remaining Archers are displayed at lower right.

The listing is pretty long, so save a copy before running it (as you should always do anyway) in case a typo error in the data statements should cause the interrupt to crash the computer.

There are three songs that play in the background during the game, tentatively titled "Archer's Theme" (title screen), "The Troll's Dirge" (dungeon overview) and "Archer's Crusade" (inside rooms). You may recognize the flute tune that plays behind the title screen–it's lifted from a melody by Mozart. ☐ *SEE PROGRAM LISTING ON PAGE 93*

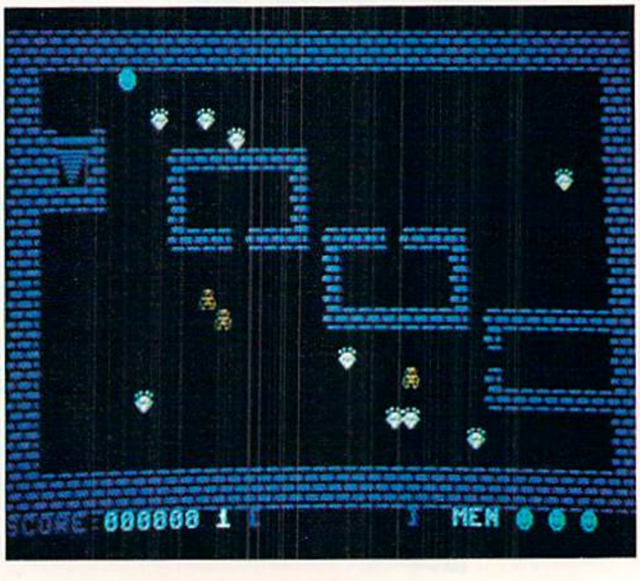


PHOTO: MICHAEL R. DAVILA

SCUTTLEBUTT

Continued from page 12

included here will procure you a copy of *Planetfall* for $14.95.

Infocom, Inc., 617-492-6000 (see address list, page 12).

JFORTH

Based on the Forth '83 standard and supporting FIG and Forth-79 as well, *JForth* for the Amiga ($99.95) contains both an interpreter and a compiler, providing an interactive environment that



It's an interpreter. It's a compiler. READER SERVICE NO. 169

makes it possible to compile programs directly from the keyboard. A technique known as JSR-Threading ties together compiled programs, making all code produced by the compiler directly executable.

Delta Research, 415-485-6867 (see address list, page 12).

BUYER BEWARE

Bad news about Schneider Software, whose offering of public domain disks was publicized in March's *Scuttlebutt*. We've heard from numerous readers who sent the $2.00 Schneider requested for a disk-based catalog, received back their cancelled checks, but...you guessed it. No disks. And no answers to the letters we've written to Schneider trying to redress the problem. If you've lost a deuce to these big-time operators, write to the Pennsylvania Attorney General's Bureau of Consumer Protection at 358 State Office Building, 100 Lackawanna Avenue, Scranton, PA 18503 (phone: 317-963-4913). If you won't do it for the two dollars, do it to help keep these folks from trying for similar mail order bonanzas in the future.

RUPERT REPORT

Continued from page 40

computer to be run in FAST or SLOW mode with the 80 column display. Upon returning to 80 columns from 40 columns, the computer will always be in SLOW mode.

USING IT

You may assemble this program yourself if you want to modify or relocate it. You may change the ORG statement in line 111. You may run the BASIC program SCRNSWAP LOADER on page 84 if you prefer. This BASIC program POKEs the machine language object code into memory, then performs a SYS 5376 to run the initialization portion.

Be sure to save the BASIC program before running it. If it gives a DATA ERROR message, carefully check your data statements and correct any errors. Resave the program after any changes.

One other possibility is to add line 255 to SCRNSWAP LOADER:

```
255 BSAVE "SCRNSWAP.O", P5376 TO P5523
```

This will create an object file called SCRNSWAP.O on the disk. If you have to correct and rerun the BASIC program, you must first scratch the old version of SCRNSWAP.O each time. Type SCRATCH "SCRNSWAP.O" before rerunning the program.

If you added line 255, then the next time you start your computer, you can load and execute the program with the statements

```
BLOAD "SCRNSWAP.O" : SYS 5376
```

This is faster than running the original BASIC loader program. The real advantage is that these statements can be part of your autoboot program. (You *do* have an autoboot program to load your favorite utilities, set your screen colors, and customize your function keys, don't you?)

CHANGING MODES

How is the screen mode changed? There are several ways to switch between the 40 column and the 80 column displays. The Escape-X sequence toggles between the two modes. Press and release the Escape key. Then press the X key.

Another way to change modes is to enter or execute the GRAPHIC command. GRAPHIC 5 selects 80-column mode. GRAPHIC followed by numbers from 0 through 4 selects the VIC 40 column output. Refer to the *Programmer's Reference Guide* for details on this statement.

Either the Escape-X or the GRAPHIC commands may be used within a program as well as in direct mode. To execute Escape-X, use PRINT CHR$(27)"X" since CHR$(27) is the code for "Escape".

Another way to change screen modes is to press the 40/80 DISPLAY key, then perform a warm restart (hold RUN STOP and press RESTORE). The 40/80 DISPLAY key is read only during warm or cold bootup. If the key is down during bootup, 80 column mode is selected. Otherwise 40 column mode is selected.

Entering GO64 and responding Y to the prompt will put the computer into C-64 mode. This automatically selects the 40 column output.

To disable the SCRNSWAP routine and any other resident IRQ or SYSTEM_VECTOR wedges, enter this as one line in direct mode:

```
POKE 2560,3 : POKE 2561,64
```

then press RUN STOP/RESTORE. These statements replace the original vector at $A00/$A01. The reboot then restores the IRQ vector. To reenable SCRNSWAP, you could enter SYS DEC("150A").

Hopefully when you write your next IRQ wedge, you will use the techniques we have discussed so that your routine does not clobber any other wedges which are present.

Anyone with a hot soldering iron and some perseverance should be able to make this project work. Write and tell me of your success or problems with it. Also let me know if you like hardware and software articles such as this.

If you are not already taking advantage of 80 column mode on the C-128, you should consider an inexpensive green screen. The 80 column mode makes the C-128 a serious programmer's computer. Electronically controlled screen swapping makes the C-128 even better. □

SEE PROGRAM LISTINGS ON PAGE 83

FSD-2 EXCELERATOR PLUS
Emerald Component International
Commodore 64
Price: $159.00

Clone is the current buzzword in the microcomputer industry. It has reached the point where the announcement of a new line of microcomputers from a certain well-known manufacturer is immediately greeted by massive speculation as to how soon the new machines will be cloned. On the other hand, the microcomputers manufactured by Commodore have never been cloned. This phenomenon may be easily explained. Commodore's microcomputers are so reasonably priced to begin with that the clonemongers have no margin for profit if they try to undercut Commodore's prices. Furthermore, the extensive use of proprietary hardware and software makes it virtually impossible to legally construct a properly functioning clone of a Commodore micro such as the C-64.

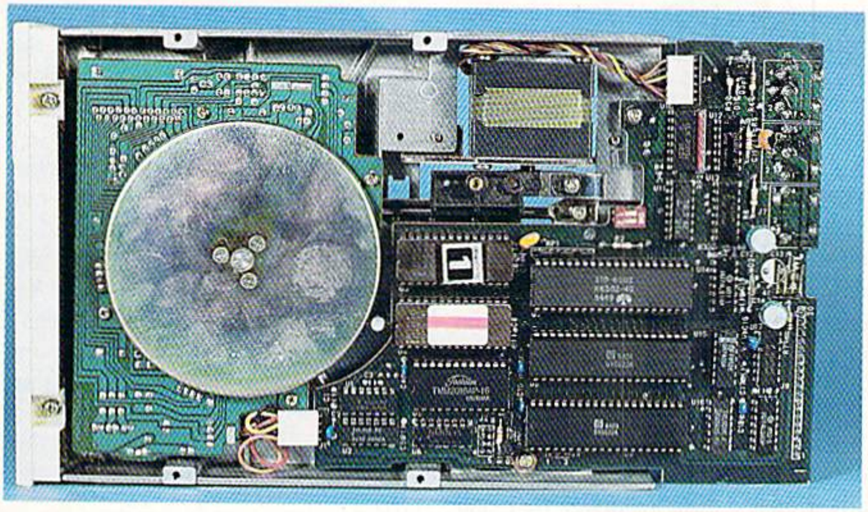
On the other hand, we have found that Commodore peripherals have been cloned from the very start. The reason for this is that for the most part Commodore does not manufacture its own peripherals. Items such as printers and disk drives are made for Commodore by major manufacturers of such products. Thus it is possible for a clone maker even to contact the original contractor for the Commodore peripheral and come to market with a competing product.

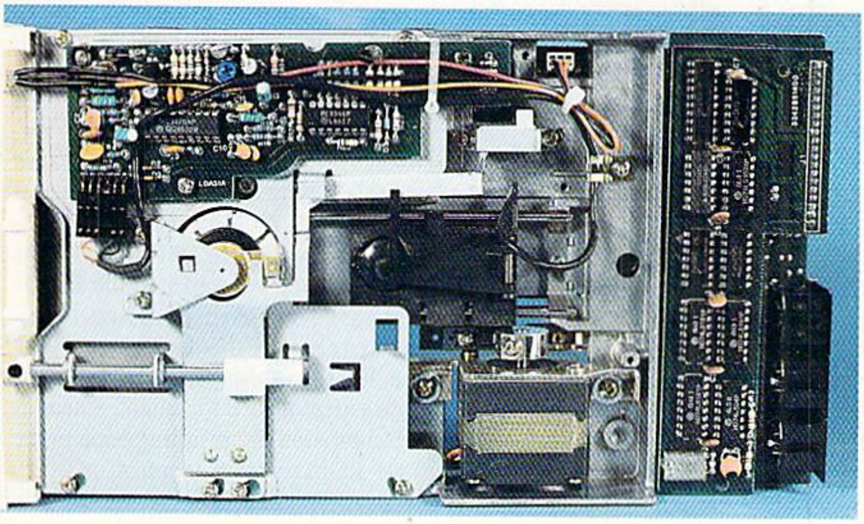
The 1541 disk drive in particular has received much attention from the clone makers. The basic hardware of the disk drive mechanism has been fairly well standardized for several years. Double sided, double density drive modules for the IBM PCs sell for well under $100. Single sided, single density modules, such as for the 1541, are even less. As a result it is very easy for a manufacturer to put together a 1541-compatible disk drive.

Distributors have also been very swift to capitalize on the 1541's reputation for misalignment under the duress of some copy protection schemes. Retailers in particular are interested in



Top: FSD-2 and its power supply atop the 1541. Middle: bottom view, showing direct drive TDK module, 6502 microprocessor, two 6522 VIAs, 16K of ROM, 2K of RAM, and DIP switches. Bottom: top view, showing remaining circuits. ***READER SERVICE NO. 167***





PHOTOS: MORTON KEVELSON

switching prospective customers to 1541 clones, as they generally have a higher markup.

The FSD-2 Excelerator Plus is another entry in the 1541 clone market. Its most distinguishing feature is its small size. The dimensions are a mere 5¾" wide by 1¾" high and 10¼" deep. This is only about one fourth the volume occupied by a 1541 disk drive. In all fairness, the additional bulk of the Excelerator Plus's external power supply should be considered as well. This sealed module, roughly the size of a C-64 power supply, adds about 30 percent to the volume of the disk drive. Even at that, the FSD-2 is less than half of the total volume of the 1541. Of course, the power supply's five foot long input and output cables let you find a convenient location for it.

The version of the Excelerator Plus we looked at is built around a half-height, direct drive TDK disk drive module. The main circuit board contains the usual collection of 1541 components consisting of a 6502 microprocessor, a pair of 6522 VIAs, 2K of RAM, and the operating system in 16K of ROM. The remainder of the circuitry is made up of standard TTL integrated circuits. To achieve its compact construction, the main circuit board has been divided up into three interconnected modules. When we opened the drive up we noticed that some of the connecting plugs were coming loose. At this point we made sure that all the connectors were properly seated.

As with similar lever action disk door designs, the disk has to be manually extracted from the drive. We found that the disk well was rather deep, making it slightly awkward to extract the disk. The front panel sports a single LED which glows green to indicate power on and red to indicate drive activity. The TDK drive mechanism has a mechanical track one stop, similar to the original 1541 drives. As expected, the drive head rattled when formatting a disk or upon encountering a disk error. The acoustics of the compact metal case minimized the sound coming from the drive during these activities. The drive's device number may be easily changed via a pair of miniature switches which are accessed through a cutout in the bottom of the drive.

Operating speed of the Excelerator Plus was comparable to the 1541 disk drive. The only notable speed improvement was in formatting a disk. The table gives some sample times in seconds.

Operation	FSD-2	1541
Format	56	85
Load 10K	26	26
Save 10K	30	31

The real concern of the end user with regard to 1541 clones is compatibility. The problems with compatibility are not with the standard Commodore DOS commands but with the nonstandard operations performed by many purveyors of commercial software. Many of these packages employ copy protection schemes which may not work on non-Commodore disk drives. The package label for the FSD-2 indicates "Guaranteed Compatibility"; however, the enclosed warranty card does not indicate the nature of this guarantee. Due to the large number of commercially protected C-64 software programs, we were unable to try out the FSD-2 with all of them, so we picked the ones we felt would be most likely to lead to difficulties.

We found no compatibility problems with the FSD-2 when we tried it out with *GEOS*, Epyx *Fastload*, and a game from Accolade Software. We even had no problems with *Super Kit 1541*, well-known for its finicky behavior with 1541 disk drives which are only slightly out of alignment. All this compatibility was truly astounding. So astounding, in fact, that we took the trouble to take a closer look at the FSD-2.

As a first step we extracted the drives ROMs, slipped them into a Promenade, and compared them to the contents of a 1541's ROM set. We were amazed to find that the code was entirely different. We were even more amazed when we noticed that the contents of the FSD-2's ROMs would not disassemble into standard 6502 opcodes. As a second step we replaced the ROMs and used the DOS's memory-read command to extract the entire contents of the FSD-2 ROMs. At this point our amazement turned to enlightenment as we compared the resulting code to the 1541 ROMs. It was identical in every respect! At this point we were ready to guarantee the compatibility of the FSD-2 ourselves.

Apparently one or more of the ROM data lines on the FSD-2's circuit board have been swapped. The ROM code was adjusted accordingly. As a result the code looks fine to the drive's 6502 microprocessor. However, a casual inspection of the ROMs will reveal only gibberish. We have been informed by a reliable source that Commodore has obtained an injunction to prevent the distribution of the FSD-2 in Germany. The presence of the FSD-2 at the last World of Commodore was also blocked. In this country the copyright status of the 1541's ROM seems to be undecided. The possibility exists that Emerald could be able to legally distribute the FSD-2 in the US.

Overall the FSD-2 Excelerator Plus is a competitively priced albeit plagiarized alternative to the Commodore 1541 disk drive. It will be especially appealing to users who will benefit from the small space occupied by the drive. It is certainly a worthwhile consideration as a second disk drive.

Emerald Components International, 541 Willamette, Eugene, OR 97401 (phone: 800-356-5178/503-683-1154).

–Morton Kevelson

GRAPHICS TRANSFORMER

Complete Data Automation, Inc.
Commodore 64
Price: $34.95

Last month we presented a roundup of graphics conversion packages along with a tutorial on bit mapped graphics for the C-64. No sooner had the ink dried on our report than another graphics conversion package arrived on our doorstep. While we suppose we should be used to this, as it seems to

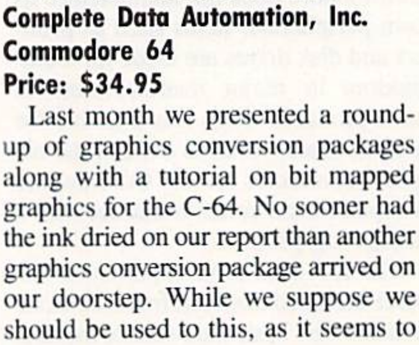


PHOTO: MORTON KEVELSON

Fills the bill for bit map maestros.
READER SERVICE NO. 168

happen often enough, somehow we are always surprised by the many devious ways of the imp of perversity. We wouldn't take the issue to heart if the package were a real dog and not worth considering in any event. However, the *Graphics Transformer* turned out to be a contender for the top spot on our list with some powers and abilities not found in any of the others.

If nothing else, the *Graphics Transformer* recognizes a greater selection of full bit map formats than any other package, as is readily apparent from the following list. You will notice that the icon formats, such as those used by *Print Shop* and *Newsroom*, are conspicuous by their absence. We have been informed that a future update to the package will incorporate these formats.

***Graphics Transformer* File Formats**

High-resolution	**Multicolor**
Art Studio	Animation Station
Billboard Maker	Billboard Maker
Cadpak 64	Blazing Paddles
Computer Eyes	Flying Colors
DOODLE!	Graphics Magician Jr.
Flexidraw	Koala
GEOS	Kwik Paint
Print Shop	Micro-Illustrator
Sahara Abstraction	Paint Magic
Vidcom 64	Peripheral Vision
Stand Alone	Picasso's Revenge
Multi File	Sorcerer's Apprentice
	Super Sketch
	Vidcom 64
	Stand Alone
	Multi File

The *Graphics Transformer* will convert any of these file formats to any other of the listed file formats. This includes the cross conversion from multicolor to high-resolution bit mapped graphics and vice versa. Note that the *Graphics Transformer* was the only package that would handle a full GEOS bit map, which may actually be as large as eight standard sized C-64 bit maps. The *Graphics Transformer* automatically loads in a separate routine to handle the conversion of GEOS files. All the other conversion functions reside in a single program module. When converting from GEOS you will be given the opportunity to scroll around the entire GEOS file and pick out a standard 8000-byte bit map.

The last two file formats under each heading are unique to the *Graphics Transformer*. The standalone format creates a file which may be loaded and then run to display the image. You have the option of leaving the image on the screen or returning to the text display mode when exiting by pressing the stop key. The Multi File format lets you save the bit map image components as separate files. For high-resolution images, these will consist of an 8000 byte bit map whose file name ends with .BIT and 1000 bytes of screen memory whose file name ends with .SCN. Multicolor images will also generate an additional 1000 bytes of color memory whose file name ends with .COL. The byte holding the multicolor background color nybble isn't retained.

The *Graphics Transformer* is entirely menu driven. Your initial task is to pick the format of the source and destination files. This drops you into the working menu where the actual conversions take place. Files are selected from onscreen directory listings. Any number of files may be selected for conversion. If more than one file is selected, they will be done in sequence. If you do not care for the results of a single conversion, you have the chance to skip the file save function for that image. However, there is no way to exit back to the main menu without loading in all the preselected files.

The directory listing may consist of files which are only in the selected source file format, or it may contain all the files on the disk. We found that the source file format listing would miss many of the *Koala* files which were actually on the disk. This forced us to use the complete directory listing to access all the *Koala* files. Selecting files from the complete directory listing lets the *Graphics Transformer* read in any disk file and treat it as the selected source file format. This would let you convert *Koala* format files which do not adhere to the *Koala* file naming convention.

When doing a multicolor to high resolution conversion or vice versa, the source file is displayed in the destination file format upon loading. We found that it was not possible to view the file in its original format without some loss of information.

The conversion is not fully automated. For most files the user must execute the Equalize command to complete the conversion. The Equalize command performs several operations depending on the conversion. When going from high-resolution to multicolor mode, Equalize will adjust the pixel bit pairs to point to screen RAM. Remember, only the 01 and 10 bit pairs will cause the VIC chip to look at screen RAM for color data. Bit pairs 00 and 11 get their color data from the background nybble and color memory. These areas are filled with default values, as they are not part of the original high-resolution file.

When going from multicolor to high-resolution mode, Equalize transfers the two most popular colors, from the available four, into the corresponding character cell of screen RAM. It also doubles up the pixels to insure that the hi-res image corresponds to the original multicolor as much as possible.

In addition to the basic format conversion, the bit map may be manipulated in several ways. All pixels may be

inverted, color may be stripped, the image may be flipped horizontally or vertically, and the bit map may be scrolled in single pixel increments. Bit map data which scrolls off the screen is lost and the color data does not scroll along with the bit map. The image may also be converted to the opposite bit map format even when the destination file format is the same as the original. The original image may be completely restored after any number of these alterations, with the exception which we noted above. Interestingly enough, there was no way to load in a file, perform some bit map manipulation, and resave the file in the same format as the original. Conversion to another format is always required.

Overall the *Graphics Transformer* is a very useful tool for anyone who works with C-64 graphics in several file formats. Many, if not all, of our criticisms will most likely be taken care of in the next update of the program. It seems that many of the proprietors of Complete Data Automation, Inc. are graduates from the select school of C-64 copy program programmers. Those of you who remember programs such as *Di-Sector* will recognize some of the names in the accompanying credits. In fact, more than a small amount of the C-64 hacker's wit has found its way into the program and its brief documentation.

Complete Data Automation, P.O. Box 1052, Yreka, CA 96097 (phone: 916-842-3431). *—Morton Kevelson*

CELEBRITY COOKBOOK VOLUME I
Merrill Ward
Commodore 64
Disk; $29.99

Here's a tasty recipe for any computerist with a hunger for high tech delicacies. Take one Commodore 64/128, add *Celebrity Cookbook Volume I* and *voilà!* One hearty appetite will surely emerge! The program might not turn you into a master chef, but it does provide alternatives to Wednesday's meat loaf and mashed potatoes. Best of all, it's designed for ease of use by noncomputerists. Simple onscreen directions, single-key commands, and full printouts of the recipes make *Celebrity Cookbook* a snap to use.

The program features 50 or more recipes collected from such varied luminaries as Ronald Reagan, Princess Di, Dolly Parton, and Bob Hope. Each recipe is presented in plain, matter-of-fact terms, with simple directions for its preparation.

As the program loads, a quote by Andy Rooney offers the pithy wisdom that "The two biggest sellers in any bookstore are cookbooks and diet books. The cookbooks tell you how to prepare the food, and the diets tell you how not to eat any of it."

The title page asks the user to choose from Volumes 1 through 6. If the cook is using one of the additional collections in this series, this is the signal to insert the disk.

After the cook chooses Volume 1, the onscreen menu lists the contents of the cookbook. There are eight choices: Appetizers & Beverages, Fish & Fowl Entrees, Meat Entrees, Desserts, Wine Directory, Bar Guide, Recipes by the Computer Chef, and Other Features. Choosing one of the four food categories prouces a listing from A to K of "star recipes." These explore the palates of the people who contributed to the cookbook, with such recipes as Ronald Reagan's Corned Beef Hash, Mitzi Gaynor's Hungarian Goulash, Jane Fonda's Curried Chicken, and Johnny Mathis' Wild Duck.

The Wine Directory is especially nice for meal planners. First the user chooses from five food categories: fowl, seafood, red meat entrees, desserts, and cheeses, nuts, and pasta. After the initial selection, the cook can further define the parameters of the meal. Under seafood, for example, the menu lists A, clams, oysters and nonoily fish, grilled or poached; B, smoked salmon, fish with cream sauces, and fried fish; or C, rich preparations or crab and lobster. Each of these selections produces a brief description of the quality of wine that best compliments the food to be served, followed by a list of a half-dozen or more specific suggestions.

The Bar Guide teaches the host or hostess the basics of mixology. There are recipes and instructions for a dozen of the most popular drinks, plus information on the amount of alcohol needed for parties from four to forty guests.

Celebrity Cookbook, in addition to the recipes, contains Star Diets, in which 20 contributors like Mary Tyler Moore, Sophia Loren, and Boy George describe what they eat for breakfast, lunch, and dinner. Most have a weight loss theme, and there are some helpful hints here for dieters. However, some are charming and ingenious. For example, Dolly Parton admits her breakfast "...used to be spare ribs. Now it's juice and a soft-boiled or poached egg."

Other features make the program even more useful. Perle Mesta's Party Tips provides hints for more pleasurable social gatherings, and includes a quiz on being a good guest. A slide show flashes a half-dozen food-connected cartoons onscreen in a meaningless but pretty display. Finally, *Celebrity Cookbook* contains an easy-to-use recipe filer. The program even comes with a formatted blank disk, already titled 'My Favorite Recipes,' for the computerist's use. To make the program more attractive, special options change the text, background screen, or screen border colors at will.

Although the onscreen instructions are fine, it is a pity that Merrill Ward didn't include printed documentation as well. It would be helpful to have information at hand, particularly when using the recipe filer. However, the program does permit the user to make printouts of the instructions, and the package includes a Quick Reference Card with a listing of command keys.

Merrill Ward promises five additional volumes to follow, at $14.99 each. Four of these will feature menus for holidays (Easter, Fourth of July, Thanksgiving and Christmas), and the fifth will have prizewinners from a contest sponsored by the publisher.

What *Celebrity Cookbook Volume I* lacks in fancy ornamentation is made up by ease of operation. It may not make you a better cook, but it should give you some new ideas about what to fix for supper tonight!

Merrill Ward, 255 N. El Cielo Road, Suite 222, Palm Springs, CA 92262 (phone: 619-320-5828).

—Joyce Worley

CCSZ CLOCK/CALENDAR CARTRIDGE
Jason-Ranheim
Commodore 64 and 128
Price: $49.95

Does your C-64 or C-128 ever want

to know the date and time? More accurately, are programs you run on your computer interested in the date and time? If so, the CCSZ Clock/Calendar Cartridge may be just what you need. Now, some of you may be aware that the TI and TI$ variables in Commodore BASIC will return the time or a reasonable facsimile thereof. And you may justly question the need for an additional time gizmo to be added to the computer's expansion port. Rest assured that we will properly address all these concerns.

BASIC does maintain a rudimentary timekeeping function. Locations 160-162 ($A0-$A2) are the zero page storage locations for the C-64's and the C-128's software jiffy clock. Note that in this case a jiffy is 1/60 of a second. This 24 hour clock is updated every 1/60 of a second by the operating system at the occurrence of each system IRQ. The system IRQ is what normally performs BASIC's housekeeping functions, which include the scanning of the keyboard. The problem with this clock is that it is not very accurate. There are numerous operations which interrupt the regular functioning of the IRQ, such as disk and tape operations.

Interestingly enough, the C-64 and the C-128 have two genuine, real time, time of day (TOD) hardware clocks built right in. These clocks are an integral part of 6526 Complex Interface Adaptor (CIA) chips, of which there are two in every computer. These are 12 hour clocks with a precision of 1/10 of a second. A single bit is provided to indicate AM or PM. The clocks are updated every 1/10 of a second by the 60 Hz (50 Hz in Europe) power line frequency whose rate is accurately maintained by the power company. This is one of the uses for the nine volt AC supplied to the computer by its external power supply.

Strangely enough, the operating system makes absolutely no use of what seems to be a very valuable hardware resource. The reason for this is probably a holdover from the PET and VIC 20, and the BASIC 2.0 which is common to all these computers. The I/O functions in these older machines were implemented with 6522 Versatile Interface Adaptor (VIA) chips. The VIA chip does not have the internal TOD clock provided with the CIA. This does not explain why a real time clock function was not added to BASIC 7.0 in the C-128, especially in view of the fact that the CIA's TOD clock is not affected by anything the computer does as long as it is turned on.

The CCSZ Clock/Calendar cartridge seeks to rectify this oversight and proceeds to do so with a vengeance. Jason-Ranheim has managed to cram an impressive array of hardware into a standard size (2½" by 3") expansion port cartridge. To start with, there is an OKI Semiconductor MSM6242RS crystal controlled clock chip. To keep this clock running after the computer is turned off there is also a 3.6 volt, 70 milliampere hour, rechargeable nickel-cadmium battery. This battery, when fully charged by approximately 24 hours of computer operation, is capable of maintaining the contents of the CCSZ for up to 18 months.

The program which tells the CCSZ what to do is stored in an onboard 8K PROM. In addition the CCSZ contains 8K of static RAM of which one page (256 bytes) is reserved by the CCSZ. The remaining 7936 bytes is available to the user. The contents of this RAM is also preserved by the CCSZ's built-in battery. Finally there is an empty 28 pin DIP socket which can accommodate a user-programmed PROM with up to 64K of program space. The emphasis on PROMs in the CCSZ is not at all surprising in view of Jason-Ranheim's flagship product, the Promenade, a very versatile PROM programmer for the C-64 priced at less than $100.

This impressive collection of chips would not be of much use without proper operating system support. This support is exactly the function of the CCSZ's built-in PROM. Incidentally, all the CCSZ's chips, except the clock chip, are socketed for easy replacement. The first time you turn on the computer with a CCSZ installed you should do so with the CONTROL key depressed. This will cause the display of the CCSZ's setting menu.

The first menu item is the Set Date and Time Function, which drops you into a second menu to do just that. Once set, the information is stored in the CCSZ's own RAM. The second menu item tells the CCSZ just what to do with its time information every time

it powers up. You can instruct the CCSZ to install the time into the TOD clock of one of the computer's CIA chips. Or you can tell the CCSZ not to update the CIA's TOD clock at all.

The next menu item tells the CCSZ the power line frequency. For North America this will be 60 Hz and for Europe it will be 50 Hz. The frequency setting in the computer is a hardware function which is not available to the CCSZ on the expansion port.

Next you are offered the option of automatically displaying the date and time information in the upper right hand corner of the screen. This information will be updated by the CCSZ's own IRQ which occurs once per second. Thus, even if you clear the screen, the date and time information will reappear. You have the option of displaying only the time, the date and the time, or nothing at all. Note that the date and time display data will be read by the computer when you hit the RETURN key on that line. This will interfere with user input on the first screen line.

The next option allows the CCSZ to date and time stamp disk files. This does not use any normally accessed space on the disk, as four unused bytes in the disk directory have been chosen for this purpose. The disk file data and time information may be displayed by using the CCSZ's directory display command, the asterisk (*). This command supports all four disk drive device numbers (8, 9, 10, or 11) and may be accessed within the C-128's MLM.

The next two functions let the CCSZ automatically load and run a user-stored program from either its own 8K RAM or from the disk. The CCSZ RAM is treated as device 12. You can store only one program in it. You can also store BASIC's variable, array, and string data in the CCSZ's RAM. Recalling BASIC's data from CCSZ RAM can be a very fast way of initializing a program.

If you elect to load a program from disk, four more options are available. These let you choose between a C-64 program or a C-128 program and between a relocatable BASIC program LOAD or a non-relocatable LOAD. You will also indicate the program's name.

All the information entered during the setup process is stored in the CCSZ's battery-backed RAM and will be implemented each time the computer is turned on or reset.

In addition to all the setup options, the CCSZ may be directly controlled via a set of 24 additional functions. These are all accessed by first POKEing a value (0-31) into 165 and then executing a SYS 999. The code which controls the function access consists of 27 bytes starting at 996 ($03E4) in both the C-64 and the C-128. We do not have the space to go into each of these routines in detail, but we have listed them all here:

Table of CCSZ Functions

No.	Function
0	Set default computer to C-64 or C-128
1	Read date and time from CCSZ and store in 166-173
2	Store BASIC variables in CCSZ RAM
3	Recall BASIC variables from CCSZ RAM
4	Store BASIC array data in CCSZ RAM
5	Recall BASIC array data from CCSZ RAM
6	Store BASIC string data in CCSZ RAM
7	Recall BASIC string data from CCSZ RAM
8	Store CCSZ bytes free in 166-167
9	Set RAM protect pointer in CCSZ RAM
10	Read power-off log from CCSZ
11	Read power-on log from CCSZ
12	Set BASIC's TI$ to current time
13	Read a register from the CCSZ's 6242 clock chip
14	Store data in 6242 register
15	Not defined
16	Store single byte to CCSZ RAM from 166
17	Read a byte from CCSZ RAM into 166
18	Store a block to CCSZ RAM
19	Read a block from CCSZ RAM
20-23	Not defined
24	Number of times CCSZ has been powered up or reset
25-27	Not defined
28	Initialize (format) CCSZ RAM
29	Completely disable CCSZ
30	Partial disable of CCSZ
31	Reenable CCSZ after function 30

When the CCSZ is installed, the auto-boot feature of the C-128 is effectively disabled. That is, the auto-boot function involving sector 0 of track 1 on the disk, which is automatically executed as part of the C-128's power up sequence, will not work. BASIC 7.0's BOOT command, which does the same thing, does not work either. The CCSZ function number 29 will disable the CCSZ and allow the BOOT command to operate. It is possible to restore the auto-boot function without removing the CCSZ from the cartridge port by including the execution of function 29 in a short program saved to those disks you wish to auto-boot. This could be made part of the program which is automatically executed by the CCSZ on power up. Based on the available CCSZ startup scenarios, there are many other ways to accomplish this task.

Functions 10, 11, and 24 are intended for when the computer is used for the control of unattended processes. These functions provide the application the ability to determine things such as the date and time of a power interruption, its restoration, and the number of operations which have occurred. When combined with the CCSZ's various autobooting capabilities, it is possible to design a computerized application which can continue to function through disturbances.

Functions 13 and 14 deal directly with the registers in the 6242 clock chip. A table of these 16 registers is included with the CCSZ manual.

The use of the CCSZ with currently available commercial applications will be spotty. Even applications which make use of the CIA's built-in TOD clock are apt to reset its contents when starting up. For example, CP/M uses the TOD clock in CIA #1 for its date function. We found that CP/M would preserve the minutes and seconds values it found there, but the hour value gets clobbered. In fact, CP/M even clobbered the hour value in the CCSZ. Note that the CIA chip does not store any date information. This data is purely a function of the CCSZ.

It may be possible to modify some applications to work with CCSZ. Telecommunications programs are a specific example which stand to benefit from the CCSZ's time and date data. Jason-Ranheim has succeeded in modifying *Bob's Term Pro 128, Sixth Sense 128,* and *Sixth Sense 64* to work with the CCSZ. The procedure for doing this is available from Jason-Ranheim upon request.

The CCSZ is not intended for everyone. Its application will tend to be somewhat specialized and it is obviously geared to users with programming experience. Those users who have a need for the CCSZ's specific capabilities will find it to be a very capable and extremely well thought out product.

Jason-Ranheim, 1805 Industrial Drive, Auburn, CA 95603 (phone: 800-421-7731; in CA 800-421-7748).

—Morton Kevelson

COMAL COLUMN

THE DEATH OF GOTO

By Richard Herring

Over the last two months we've talked about control structures – commands that break normal line-by-line program execution and transfer the path of execution to some other point in the program. We covered all the control structures in COMAL (and in most other languages) except for two.

The best and the worst were saved for last. The best? Procedures (PROC/ENDPROC). The worst? GOTO. Most of us started in BASIC. That means we were introduced to the worst, most misused control structure and deprived of the best.

In a classic letter to the editor, Edsger Dijkstra (a veritable philosopher of programming) observed that the ease of understanding program listings is inversely proportional to the number of unconditional transfers of control (GOTOs) in those listings. The editor titled the letter "Go to Statement Considered Harmful."

What Dijkstra was selling was not just the demise of GOTO, but the advent of structured programming. That's a methodological style of building computer programs by logically linking subprograms that are either themselves structured programs or are in the form of control structures.

Back in March we developed a graphics demo where the main program was composed entirely of procedure calls and was only three lines long:

```
10 questions
20 setup
30 drawit
```

Alternatively, we could have used GOTOs:

```
10 GOTO <label at line 40>
20 GOTO <label at line 210>
30 GOTO <label at line 330>
```

Then, at the end of each of our old procedures, we could have jumped back to the main program, or to another procedure, or to somewhere seemingly unrelated in the program. Unlike the control structures we studied in the last two issues, the pattern used by GOTO is neither systematic nor obvious.

GOTO says to the reader of the program, "Stop here. You'll find the continuation of this logic at some other point. Then you may come back here or you may not." The problem with GOTOs is that they fool the programmer into believing that he has control. Programs that use lots of GOTOs can continue for pages and pages, borrowing routines from themselves, without ever breaking into subprograms that will still be comprehensible to the programmer a few months down the road.

Structured programming, as an alternative, is designed to be easy to read, easy to understand, easy to update, and easy to debug. During the actual coding of the program, structured programming will actually prevent errors.

The popularity of structured programming is apparent throughout computerdom. Witness the popularity of the structured language Pascal in universities, the U.S. government's choice of a structured format for its chosen language Ada, and the acceptance of FORTRAN 77 (a version that finally allows some structure) in the scientific community.

Now BASIC doesn't prevent structured programming, but the design of the language does not encourage it either. COMAL's design does. Especially by fostering the use of procedures.

Procedures are nothing more than stand-alone program modules. Ideally they are never over a page long (unless they are broken into recognizable subunits themselves). A COMAL procedure can be called from the main program by just listing its name on a line by itself. To make reading easier, COMAL will automatically indent the body of the procedure.

When you call a procedure, the program will jump to the procedure, execute it, then jump back to the main program and continue with the line right after the procedure call. Procedures will not be run unless your program calls

them, even though they are right there in the listing.

In COMAL 2.0 procedures become especially attractive because they can be EXTERNAL to the program. You can save them separately on disk and call them from your program whenever you like. Given the speed of our faithful Commodore disk drives, you won't do this a lot. But in large programs, the ability to have a subroutine (in the guise of an EXTERNAL procedure) that only uses memory while it's in use, then gives that memory back to the main program, is pretty attractive.

Program variables are generally considered global–available to both the main program and the procedure. What if the internal operation of your procedure interferes with the main program, say by changing a value you don't want changed? No problem. Just declare the procedure CLOSED and every variable will be local–known only to the procedure, even if it has the same name as a variable in the main program.

Once a procedure is CLOSED, you can pass the value of a variable into the procedure by listing that variable in the procedure heading. Or you can IMPORT the variable into the procedure (in 2.0 only). When a variable is passed into a procedure, the main program still uses its original value. When that variable is IMPORTed, however, the main program will recognize any changes made by the procedure.

COMAL procedures, like procedures in all languages that support them, are like black boxes. They have certain inputs, which we can control, and certain outputs, which we can specify. From the program's perspective, how a procedure (black box) does its work is not really important.

The beauty of structured programs, as E. D. Reilly wrote, is that they are worthy of being read by humans, not just by machines. And they allow programming to be approached more as a science than as an art.

Whether you're a scientist or an artist, drop me a line with your COMAL comments at P.O. Box 1544, Tallahassee, FL 32302. □

Tech Notes:

1) Dijkstra's letter to the editor was published in 1968 in "Communications of the Association for Computing Machines," Comm. ACM 11, No. 3.

2) E. D. Reilly's comments on structured programming can be found in the *Encyclopedia of Computer Science and Engineering.*

3) When you include a list of parameters with a procedure, they are passed to the procedure by their position in the parameter list, not by name. That's why names don't have to match. But the calling statement and the procedure heading must have exactly the same number of parameters. And the parameters must be in the same order.

4) If you can avoid using global variables in procedures, do. They can produce unwanted side effects that ripple through your program whenever a change is made. Also, if you avoid global variables, you will have procedures that can be individually saved to disk and used in other programs. The best programmers have libraries of routines like this.

5) After a program is RUN (or SCANned in version 2.0), the computer knows all the procedures. You can run any of the procedures in direct mode by typing EXEC <procedure name>. This even holds true for external procedures.

6) You may not give a variable and a procedure the exact same name in COMAL.

7) A procedure placed within another procedure (nested) will be local to the parent procedure, not global to the main program. Nested procedures are only allowed in version 2.0. Procedures are generally not allowed inside control structures.

8) When a procedure is CLOSED, it is not only prevented from changing any variables in the main program, but also from using any procedures, labels, or variables from the main program. CLOSED means isolated. In version 0.14, procedures and functions are always global, even after a CLOSED command.

9) In principle, GOTO is sufficient for all conceivable intraprogram sequence control. In fact, it has been proved that any flowchart or program can be rewritten, in an equivalent way, using subunits of only three kinds–command sequences, decision clauses (IF/THEN), and repetition (like REPEAT/WHILE). Bohm and Jacopini, "Flow Diagrams, Turing Machines, and Languages With Only Two Formation Rules," Comm. ACM 9, No. 5, 1964.

COMMODARES
PROGRAMMING CHALLENGES
By Dale Rupert

Each month, we'll present several challenges designed to stimulate your synapses and toggle the bits in your cerebral random access memory. We invite you to send your solutions to:

Commodares, c/o *Ahoy!*
P.O. Box 723
Bethel, CT 06801

We will print and discuss the cleverest, simplest, shortest, most interesting and/or most unusual solutions. Be sure to identify the *name* and *number* of the problems you are solving. Put *your* name and address on the listings as well. Show sample runs if possible. Briefly describe your solutions and tell what makes them unique or interesting, if they are. You must enclose a stamped, self-addressed envelope if you want any of your materials returned. Solutions received by the middle of the month shown on the magazine cover are most likely to be discussed, but you may send solutions and comments any time. Your original programming problems, suggestions, and ideas are equally welcome. The best ones will become *Commodares!*

PROBLEM #44-1: ADDED DIFFICULTY

This problem was submitted by James Bickers (Shepherdsville, KY), and he says it is HARD. The user enters a positive integer and the computer prints out all possible addition problems (of positive integers) which equal that number. For example, an input of 4 yields 1+1+1+1, 1+1+2, 1+3, and 2+2. Jim says no repeats are allowed and numbers must be ordered consecutively. Who can refuse a challenge like that?

PROBLEM #44-2: FRANK'S CHILDREN

Here's another interesting one from Necah Buyukdura (Ankara, Turkey). Frank has five children:

1. Andy is older than Bill by half of Carl's age.
2. Bill is older than Emma by half of Andy's age.
3. Carl is older than Emma by two years plus twice Emma's age.
4. Dave is one year older than Emma.
5. Emma's age is one-fourth the sum of Carl's and Dave's ages.

Write a program to calculate and print the ages of Frank's children.

PROBLEM #44-3: INNER STRING

Write an unordered INSTR (instring) program. The user enters two strings. The program tells whether or not all the letters of the first string are contained within the second string regardless of order. Any duplicate letters in the first string must also be duplicates in the second.

For example, if the user enters DOOR, COMMODARES the computer responds "Yes." If the user enters ROAR, COMMODARES the computer responds "No," since the first string contains two R's and the second has only one.

PROBLEM #44-4: KEY FUNCTIONS

What is your best, shortest, longest, most interesting, most unusual, most useful, or most exciting function key definition? One of mine for the C-128 is a "list to sequential file or printer output" key. It is defined this way:

```
CHR$(27) + "IDN=8+4*(DN<>8):OPENDN,DN,DN
,(F$):CMDDN:
LIST:PRINT#DN:CLOSEDN" + CHR$(13)
```

To send a listing to the printer, simply press the function key. DN is initially 0, so it becomes 4. Logical file 4 to device 4 (the printer) is opened, and the listing is sent to it with the CMD command.

To convert a program to a sequential disk file, first enter DN=8 : F$="filename,S,W" in direct mode, then press the function key. Now DN remains 8 and device number 8 (the disk drive) is opened. The program currently in memory is then listed to the disk file whose name is given by F$. No doubt this function key definition is trivial compared to some of those you can come up with.

This month we have a group of unique and interesting solutions to *Commodares* from the April 1987 issue of *Ahoy! Problem #40-1: Small Straight* was submitted by Steven Steckler (Columbia, MD). The idea is to simulate the toss of five dice as in the game Yahtzee, and to identify any tosses which contained one or more small straights. A small straight is a sequence of numbers 1234, 2345, or 3456 arranged in any order on the dice.

Most solutions used sorting techniques to arrange the randomly chosen dice values in numerical order. Then it is a straightforward process to look for one or more of the possible small straights.

Since we will discuss sorting techniques for the next problem, let's look at solutions which did not explicitly sort the dice first.

Harry Stoddart (Bradford, ON) used the signum (SGN) function in a relevant manner.

```
·1 REM ==================================
·2 REM    COMMODARES PROBLEM #40-1 :
·3 REM       SMALL STRAIGHT
·4 REM    SOLUTION BY
·5 REM       HARRY STODDART
·6 REM ==================================
·10 PRINT"THE #'S:":FOR X=1 TO 5:R=INT(RN
D(TI)*5+.5)+1:A(R)=A(R)+1:PRINT R;:NEXT
```

```
•20 FOR Y=1 TO 3:FOR X=Y TO Y+3:S(Y)=S(Y)
+SGN(A(X)):NEXT X:NEXT Y:PRINT
•30 PRINT"THE SMALL STRAIGHTS ARE"1234*-(
S(1)=4);2345*-(S(2)=4);3456*-(S(3)=4)
•40 RUN
```

Line 10 generates five random numbers from 1 to 6 to represent the values of the dice. A(R) stores the count of dice having the value R. If there are two 5's and no 1's, then A(5) equals 2 and A(1) equals 0.

Line 20 looks for 1234, 2345, and 3456 when Y has values 1, 2, and 3 respectively. When Y is 1, X is given values 1, 2, 3, and 4. S(Y) is incremented by SGN(A(X)) for each X. If A(1) is 0, meaning that no die has a value of 1, then SGN(A(1)) is 0. If one or more dice have a value of 1, then A(1) is greater than zero, and SGN(A(1)) is 1.

Consequently to have the small straight 1234, S(1) will be exactly 4, since there will be one or more of the values 1, 2, 3, and 4. The SGN function adds only one to the sum S(Y) even if more than one die has the value Y.

Line 30 prints 0 for any of the three possible small straights for which S(Y) is not 4. It prints the values of each small straight for which S(Y) is 4. If the logical statement S(1)=4 is true, then it has the logical value −1; otherwise it has the value 0. Consequently 1234 * −(S(1)=4) equals 1234 when S(1) is 4, and it equals 0 otherwise.

A different approach is shown in this COMAL program by Mark Breault (Brandon, MAN)


```
0001 //==================================
0002 //   COMMODARES PROBLEM #40-1 :
0003 //        SMALL STRAIGHT
0004 //   SOLUTION BY
0005 //        Mark Breault
0006 //======= COMAL 0.14 =============
0010 dim d$ of 6
0020 d$:="000000"
0030 for i:=1 to 5 do
0040   r:=rnd(1,6)
0050   d$(r):="1"
0060   print r,
0070 endfor i
0080 if "1111" in d$ then print " Small
Sraight"
```

Mark uses the character positions within the string variable D$ to store the results of the dice. D$ is initialized to "000000". For every value displayed on the dice, the corresponding 0 in D$ is changed to a 1. If the roll of the dice gives values 1, 1, 2, 2, and 5, then D$ would have the value "110010" where characters in positions 1, 2, and 5 are 1's.

All that is necessary to find a small straight is to look for four consecutive 1's within D$. That is the purpose of line 80. This program inherently sorts the values of the dice and makes it easy to pick out the small straight.

BASIC 7.0 allows the MID$ statement on the left side of the equal sign. To set the Rth character within D$ to "1", line 50 of Mark's program would be

```
50 MID$(D$,R)="1"
```

and the equivalent statement for line 80 is

```
80 IF INSTR(D$,"1111") <> 0 THEN PRINT "
SMALL STRAIGHT"
```

INSTR (called "in-string") gives a number corresponding to the first occurrence of "1111" within D$. If D$ does not contain the substring "1111", then the INSTR function has the value 0.

Other variations on this theme of inherent sorting included binary and prime number representations of the dice values. Stephane Edwardson (La Tuque, QUE) assigned prime numbers 2,3,5,7,11, and 13 to dice values 1 through 6 respectively. If the dice values of a toss were 1, 4, 5, 2, and 3, the prime product of the toss would be 2 * 7 * 11 * 3 * 5 or 2310. The prime products of the small straights 1234, 2345, and 3456 are 210, 1155, and 5005 respectively.

If the roll of the dice contains any of the small straights, then its prime product will be evenly divisible by the prime product of that small straight. In the example above, 2310 is evenly divisible by 210 and by 1155 but not by 5005. Therefore that toss of the dice must contain the small straights 1234 and 2345 but not 3456. Ah, the pleasures of mathematics! Merci, Stephane.

Problem #40-2: Short Sort was suggested by Chris Raimondi (Kingsville, MD). The problem was to see if anyone could come up with a one-line sorting routine which would arrange six input values in numerical order.

The two major hurdles in fitting the routines into one line were 1) the 80-column line length limit for the C-64, and 2) the pesky IF-THEN statement without an ELSE on the C-64 (nothing on a line after the IF clause is executed if the clause is not true). Clever coding and abbreviations solved the first problem. Many readers ingeniously used logical variables to solve the second.

The following program is a compendium of a few of the multitude of one-line sorts.

```
•1 REM ====================================
•2 REM    COMMODARES PROBLEM #40-2 :
•3 REM       SHORT SORT
•4 REM ====================================
•10 FOR I=1 TO 6 : A(I)=INT(RND(0)*1000)-
500:PRINT A(I);:QQ(I)=A(I):NEXT I:PRINT
•20 FOR JJ=1 TO 5 : ON JJ GOSUB 110,210,3
10,410,510
•30 FOR I=1 TO 6 : A(I)=QQ(I) : NEXT I :
PRINT : NEXT JJ
•99 END
•100 REM * SOLUTION BY JIM SPEERS *******
****************************************
•110 FORI=1TO6:K=1:FORJ=1TO6:K=K-(A(J)>A(
K))*(J-K):NEXT:PRINTA(K);:A(K)=-9E9:NEXT
•120 RETURN
•200 REM * SOLUTION BY STEPHANE EDWARDSON
****************************************
•210 FORY=1TO5:FORZ=YTO6:A=A(Y):B=A(Z):A(
Y)=-A*(A<=B)-B*(B<A):A(Z)=A+B-A(Y):NEXTZ
,Y:FORN=1TO6:PRINTA(N);:NEXT
•220 RETURN
•300 REM *  SOLUTION BY RON GOW *********
****************************************
•310 FORX=1TO6:B=0:FORY=1TO6:B=B-(A(Y)<=A
(X)):NEXT:C(B)=A(X):NEXT:FORN=1TO6:PRINT
C(N);:NEXT
•320 RETURN
•400 REM *  SOLUTION BY D.BRUCE POWELL **
****************************************
•410 FORI=1TO5:FORJ=I+1TO6:T=A(J):D=T-A(I
):K=D<0:A(J)=T+K*D:A(I)=T-(1+K)*D:NEXTJ:
PRINTA(I);:NEXTI:PRINTA(I);
•420 RETURN
•500 REM *  SOLUTION BY JEFF ACHTNIG  ***
************* C-128 ONLY ***************
•510 FORY=1TO6:F=999999999:A(Z)=F:FORX=1T
O6:IFF>A(X)THENF=A(X):Z=X:NEXTX:PRINTF;:
NEXTY:ELSENEXTX:PRINTF;:NEXTY
•520 RETURN
```

Line 10 randomly selects six integers between −500 and 500. They are stored in A( ) and also in QQ( ). Line 20 calls each of the sort-and-print subroutines in turn. Line 30 restores the original values of A( ) before the next routine is called. Line 110 from Jim Speers (Niles, MI) replaces each element of A( ) with a very small value after that value is sorted. Jim's program sorts the elements in descending order.

The other programs sort in a ascending order. C-64 users may have to use two lines for some of these routines, although that is because of PRINT statements. The sorting portion of each of these programs fits into one 80-character line. These programs are representative of the wide variety of solutions, although there were dozens of variations on these themes.

Thanks to Stephane Edwardson (La Tuque, QUE), Ron Gow (Sacramento, CA), D. Bruce Powell (Franklin, VA), and Jeff Achtnig, as well as all others who sent short sorting solutions.

An interesting approach for quickly sorting long lists of integers was described by Walter Deuchler, Jr. (Aurora, IL). Rather than store the integers in an array, you can store a count of the number of occurrences of each integer in the array. The index of the array is the integer value.

Say that you have a set of three integers ranging from 0 to 99. If the numbers are 5, 23, and 23, then A(5)=1, A(23)=2, and all other elements of A( ) from A(0) to A(99) are zero. To print the set in sorted order, simply step through each element of A( ) and print the index of each element the appropriate number of times. Look at Walter's program to see how it's done.

```
•1 REM ====================================
•2 REM    COMMODARES PROBLEM #40-2 :
•3 REM       SHORT SORT
•4 REM    SOLUTION BY
```

```
•5 REM         WALTER DEUCHLER JR.
•6 REM ====================================
•10 MX=99 : DIM A(MX) :REM MAX. INTEGER
•20 NM=10             :REM # TERMS
•30 FOR J=1 TO NM : X = INT(RND(0)*100) :
 A(X)=A(X)+1 : NEXT
•40 FOR J=0 TO MX : T=A(J) :IF T THEN FOR
 K=1 TO T : PRINT J; : NEXT K
•50 NEXT J
```

The price you pay for the speed of this sorting routine is the amount of memory used. The array A( ) must have as many elements as the largest allowable integer. Line 10 sets the maximum integer to be 99 and dimensions A( ). Line 20 sets NM to the number of items to be sorted. Line 30 picks NM random integers from 0 to 99. If a 5 is picked, for example, then one is added to A(5). As the numbers are picked, they are automatically sorted.

Line 40 prints the selected integers in sorted order. It looks at all elements of A(J) from A(0) to A(99). If one 0, no 1's, and three 2's had been picked, then A(0)=1, A(1)=0, and A(2)=3. Consequently the K loop in line 40 prints 0, 2, 2, 2. The statement "IF T THEN" tests the value of T. If T is zero, the rest of line 40 is skipped. If T is non-zero, the K loop prints the value of J, T times.

Walter mentioned that he has sorted an array of 100 integers in 304 jiffies using this "indexing" sort, whereas the Quicksort required 1884 and the Shell sort required 3128 jiffies. Those last two are well-known sorting algorithms. If your data fits the constraints of this "indexing" sort, you can't beat it for speed and ease of implementation. By the way, Robert Sedgewick in his very useful book *Algorithms* (Addison-Wesley, 1984) calls this type of sort a "distribution counting."

Ivan Rudyk sent the following implementation of the Quicksort.

```
•1 REM ====================================
•2 REM      DEMONSTRATION OF QUICKSORT BY
•3 REM          IVAN RUDYK
•4 REM ====================================
•10 Q=180 : DIM N(Q),M(Q)
•20 PRINT"[CLEAR]GENERATING NUMBERS[3"."]
 ":FOR X=1 TO Q:N(X)=INT(1000*RND(0))+1:N
EXT X
•30 PRINT"SORTING[3"."]" : T0=TI
•40 M(1)=1 : M(2)=Q : A=2
•50 B=M(A) : A=A-1 : C=M(A) : A=A-1 : E=C
•60 F=B : D=N((C+B)/2)
•70 IF N(E)<D THEN E=E+1 : GOTO 70
•80 IF N(F)>D THEN F=F-1 : GOTO 80
•90 IF E<=F THEN Z=N(E) : N(E)=N(F) : N(F
)=Z : E=E+1 : F=F-1
•100 IF E<=F THEN 70
•110 IF C<F THEN A=A+1 : M(A)=C : A=A+1 :
 M(A)=F
•120 C=E : IF C<B THEN 60
•130 IF A<>0 THEN 50
•140 FOR X=1 TO Q : PRINT N(X); : NEXT X
•145 PRINT TI-T0"JIFFIES" : END
•200 REM ==================================
•210 REM       CONVENTIONAL BUBBLE SORT
•220 REM        (ADD LINE 35 GOTO 240)
•230 REM ==================================
•240 FOR J=1 TO Q-1 :FOR K=J+1 TO Q : IF
 N(J)>N(K) THEN T=N(J) :N(J)=N(K) :N(K)=T
•250 NEXT K,J : FOR X=1 TO Q : PRINT N(X)
 ;: NEXT X : PRINT TI-T0"JIFFIES" : END
```

It generates and sorts 180 random integers from 1 to 1000. It keeps track of the number of jiffies (sixtieths of a second) and displays the results.

You may add line 35 GOTO 240 to this program to use the more traditional form of the bubble sort at the end of Ivan's program. Change the value of Q in line 10 for different numbers of items to be sorted. You will soon realize that the complexity of the Quicksort program is readily offset by its speed, at least when the number of items to be sorted is large.

The Quicksort took 1484 jiffies (under 25 seconds) to sort 180 integers on the C-128 in FAST mode. The bubble sort took much longer. Run the timing tests yourself.

Problem #40-3: Polygon Play suggested by Nolan Whitaker (Jeffersonville, KY) brought out the geometry experts. Given the number of sides of a regular convex polygon, the computer gives a) the size of each interior angle, b) the sum of its interior angles, c) the area of the polygon,



HAS EVERYTHING!

SUBSCRIBE TO AHOY!

☐ Twelve Issues for $23.00 ($30.00 Canada and Elsewhere)

☐ Twenty-four Issues for $44.00 ($55.00 Canada and Elsewhere)

Name____________________________________

Address__________________________________

City______________State______________Zip__________

Send coupon or facsimile to:
ION INTERNATIONAL INC.
45 West 34th Street, Room 500, New York, NY 10001

d) the radius of an inscribed circle, and e) the radius of a circumscribed circle, assuming each side of the polygon is one meter in length. Whew!

"Regular" means "all sides equal," and "convex" means "no interior angles greater than 180 degrees." A square is a regular convex polygon of four sides. This program from Joe Steininger (Bethlehem, PA) lets the user specify the number of sides and the length of each side.

```
•1 REM ===================================
•2 REM COMMODARES PROBLEM #40-3 :
•3 REM     POLYGON PLAY
•4 REM SOLUTION BY
•5 REM     JOE STEININGER
•6 REM ===================================
•10 PRINT"[CLEAR]HOW MANY SIDES"
•20 INPUT N:IF N<3 GOTO 20
•30 PRINT"LENGTH OF EACH SIDE, IN METERS"
•40 INPUT L
•50 A=((N-2)/N)*180
•60 S=A*N
•70 R=(L/2)*(1/TAN([PI]/N))
•80 D=(L/2)*(1/SIN([PI]/N))
•90 Y=N*(L[UPARROW]2)/4*(1/TAN([PI]/N))
•100 PRINT"INTERIOR ANGLES ARE ";A;"DEGRE
 ES EACH":PRINT
•110 PRINT"SUM OF INTERIOR ANGLES IS ";S;
 "DEGREES":PRINT
•120 PRINT"THE AREA IS ";Y;"SQUARE METERS
 ":PRINT
•130 PRINT"INSCRIBED RADIUS IS ";R;"METER
 S":PRINT
•140 PRINT"CIRCUMSCRIBED RADIUS IS ";D;"M
 ETERS":PRINT:PRINT
```

The formulas for the various parameters are in lines 50-90. Keep in mind that BASIC assumes angles are measured in radians, not degrees (pi radians = 180 degrees).

Several readers used a different formula for polygon area. If R is the radius of the inscribed circle, the area of the polygon is simply N*R/2 where N is the number of sides.

Richard van Frank (Montclair, NJ) sent a very nice analysis of this problem. He was interested in the largest number of sides the C-64 could handle. It turned out to be a 4,294,967,295-sided polygon. Do you think you could distinguish between such a figure and a circle?

The inscribed and circumscribed radii are the same for this polygon (within the resolution of the computer). You might try putting this problem into a loop and watching the inscribed and circumscribed approach each other as the number of sides increases. You might also add formulas to calculate the areas of the inner and outer circles and compare them to the area of the polygon.

On the lighter side, Jim Speers sent along this bit of polygon trivia. He found when working with this problem that a regular polygon with 5.33864716 sides has an area of exactly 2 square meters. Would anyone care to build one and measure it?

The final *Problem #40-4: Integer Functions* was to give user-defined functions FNF(X) and FNC(X) which calculate the floor and ceiling function of any input value X. The floor function of X is the greatest integer less than or equal to X. The ceiling function is the least integer greater than or equal to X.

Half of this problem was nearly trivial, and the other half was somewhat tricky. Most readers defined FNF(X) this way:

```
DEF FNF(X) = INT(X)
```

In fact the INT function is equivalent to the floor function. The trickier part was the ceiling function.

The most common solution was DEF FNC(X) = INT(X) +1 but unfortunately this is not totally correct. It is valid for all values of X *except* when X is an integer. In that case, the floor function is equivalent to the ceiling function. Jim Speers used this definition for the ceiling function:

```
DEF FNC(X) = FNF(X) + 1 + (X=INT(X))
```

The last part of the expression subtracts the added one whenever X is an integer. Stephane Edwardson took a different approach. His definition of the ceiling function is

```
DEF FNC(X) = -INT(-X)
```

Convince yourselves that these two definitions give identical results. Congratulations to Thomson Fung (San Diego, CA) for recognizing this subtlety of the ceiling function.

There should be enough work here to keep you busy and happy until next month. Keep those solutions and suggestions coming.

This is a list of people with valid solutions who were not already mentioned this month.

Kirk Adams
Sean Bates (N. Stonington, CT)
Gregg Bell (Broomfield, CO)
James Bickers
(Shepherdsville, KY)
Thomas Braun (Omaha, NE)
Leo Brenneman (Erie, PA)
Vaughn Brigham (Cadillac, MI)
David Brouse (Shippensburg, PA)
Carlos Centeno (Lares, PR)
Henry Christensen (Salem, OR)
Dan Darlington
(Minneapolis, MN)
Curt Donofrio
Peggy Eckbold (Somers Pt., NJ)
Veli-Matti Eerola
(Kerava, Finland)
Bill Garrett (Longview, WA)
Alan Gutierrez
Jim Henderson (APO NY)
David Hoffner (Brooklyn, NY)
Gary Jones (Williamsport, IN)
Ron Jordan (Florence, OR)
Chuck Lahmeyer
(Temple City, CA)
John Larry (St. Albans, VT)
Jim Lostetter (Kelseyville, CA)
Jorge Milke
(Mexico City, Mexico)
Ken Morrow (El Paso, TX)
Fred Randall (Ypsilanti, MI)
Kenneth Rath (London, ONT)
Norman Richards (Angleton, TX)
Larry Schafer (Fowler, MI)
Robert Sherer (Columbus, MS)
Troy Shoap (Shippensburg, PA)
J.H. Smalley (Boulder, CO)
G. Neil Spokes
(Briarcliff Manor, NY)
Dennis Sturdevant
(Rohnert Park, CA)
Robert Washburne
(Harrisburg, PA)
Nolan Whitaker
(Jeffersonville, KY)
Carol Wycoff (Hobbs, NM)

PROGRAM LISTINGS

Attention new *Ahoy!* readers! You must read the following information very carefully prior to typing in programs listed in *Ahoy!* Certain Commodore characters, commands, and strings of characters and commands will appear in a special format. Follow the instructions and listings guide on this page.

On the following pages you'll find several programs that you can enter on your Commodore computer. But before doing so, read this entire page carefully.

To insure clear reproductions, *Ahoy!'s* program listings are generated on a daisy wheel printer, incapable of printing the commands and graphic characters used in Commodore programs. These are therefore represented by various codes enclosed in brackets [ ]. For example: the SHIFT CLR/HOME command is represented onscreen by a heart . The code we use in our listings is [CLEAR]. The chart below lists all such codes which you'll encounter in our listings, except for one other special case.

The other special case is the COMMODORE and SHIFT characters. On the front of most keys are two symbols. The symbol on the left is obtained by pressing that key while holding down the COMMODORE key; the symbol on the right, by pressing that key while holding down the SHIFT key. COMMODORE and SHIFT characters are represented in our listings by a lower-case "s" or "c" followed by the symbol of the key you must hit. COMMODORE J, for example, is represented by [c J], and SHIFT J by [s J].

Additionally, any character that occurs more than two times in a row will be displayed by a coded listing. For example, [3 "[LEFT]"] would be 3 CuRSoR left commands in a row, [5 "[s EP]"] would be 5 SHIFTed English Pounds, and so on. Multiple blank spaces will be noted in similar fashion: e.g., 22 spaces as [22 " "].

Sometimes you'll find a program line that's too long for the computer to accept (C-64 lines are a maximum of 80 characters, or 2 screen lines long; C-128 lines, a maximum of 160 characters, 2 or 4 screen lines in 40 or 80 columns respectively). To enter these lines, refer to the *BASIC Command Abbreviations Appendix* in your User Manual.

On the next page you'll find our *Bug Repellent* programs for the C-128 and C-64. The version for your machine will help you proofread programs after typing them. (Please note: the *Bug Repellent* line codes that follow each program line, in the whited-out area, should *not* be typed in. See instructions preceding each program.)

On the second page following you will find *Flankspeed*, our ML entry program, and instructions on its use.

Call *Ahoy!* at 212-239-6089 with any problems (if busy or no answer after three rings, call 212-239-0855).

WHEN YOU SEE	IT MEANS	YOU TYPE	YOU WILL SEE	WHEN YOU SEE	IT MEANS	YOU TYPE	YOU WILL SEE
[CLEAR]	Screen Clear	SHIFT CLR/HOME		[BLACK]	Black	CNTRL 1	
[HOME]	Home	CLR/HOME		[WHITE]	White	CNTRL 2	
[UP]	Cursor Up	SHIFT ↑ CRSR ↓		[RED]	Red	CNTRL 3	
[DOWN]	Cursor Down	↑ CRSR ↓		[CYAN]	Cyan	CNTRL 4	
[LEFT]	Cursor Left	SHIFT ← CRSR →		[PURPLE]	Purple	CNTRL 5	
[RIGHT]	Cursor Right	← CRSR →		[GREEN]	Green	CNTRL 6	
[SS]	Shifted Space	SHIFT Space		[BLUE]	Blue	CNTRL 7	
[INSERT]	Insert	SHIFT INST/DEL		[YELLOW]	Yellow	CNTRL 8	
[DEL]	Delete	INST/DEL		[F1]	Function 1	F1	
[RVSON]	Reverse On	CNTRL 9		[F2]	Function 2	SHIFT F1	
[RVSOFF]	Reverse Off	CNTRL 0		[F3]	Function 3	F3	
[UPARROW]	Up Arrow	↑	↑	[F4]	Function 4	SHIFT F3	
[BACKARROW]	Back Arrow	←	←	[F5]	Function 5	F5	
[PI]	PI	π	π	[F6]	Function 6	SHIFT F5	
[EP]	English Pound	£	£	[F7]	Function 7	F7	
				[F8]	Function 8	SHIFT F7	

BUG REPELLENT FOR THE 64 & 128 By BUCK CHILDRESS

Please note: the *Bug Repellent* programs listed here are for *Ahoy!* programs published from the May 1987 issue onward! For older programs, use the older version.

Type in, save, and run *Bug Repellent.* You'll be asked if you want automatic saves to take place. If so, you're prompted for the device, DISK (D) or TAPE (T). You then pick a starting file number, 0 through 99. Next, you enter a name, up to 14 characters long. At this point, *Bug Repellent* verifies your entries and gives you a chance to change them if you want. If no changes are needed, *Bug Repellent* activates itself. (Pressing RETURN without answering the prompts defaults to disk drive and begins your files with "00BACKUP".)

As you enter program lines and press RETURN, a *Bug Repellent* code appears at the top of your screen. If it doesn't match the code in the program listing, an error exists. Correct the line and the codes will match.

If used, automatic saves take place every 15 minutes. When the RETURN key is pressed on a program line, the screen changes color to let you know that a save will begin in about three seconds. You may cancel the save by pressing the RUN STOP key. The file number increments after each save. It resets to 00 if 99 is surpassed. After saving, or cancelling, the screen returns to its original color and the timer resets for 15 minutes.

When you've finished using *Bug Repellent*, deactivate it by typing SYS 49152 [RETURN] for the Commodore 64 or SYS 4864 [RETURN] for the Commodore 128.

C-64 BUG REPELLENT

```
·10 PRINTCHR$(147)"LOADING AND CHECKING THE DATA[3"."]":J
=49152
·20 FORB=0TO11:READA:IFA<0ORA>255THEN40
·30 POKEJ+B,A:X=X+A:NEXTB:READA:IFA=XTHEN50
·40 PRINT:PRINT"ERROR IN DATA LINE:"PEEK(64)*256+PEEK(63)
:END
·50 X=0:J=J+12:IFJ<49456THEN20
·60 POKE198,0:POKE49456,0:A$="Y":B$=A$:C$="D":D$="DISK":D
=8:PRINTCHR$(147)
·70 INPUT"DO YOU WANT AUTOMATIC SAVES (Y/N)";A$:PRINT:IFA
$="Y"THEN90
·80 PRINT"NO AUTOMATIC SAVES[3"."]":GOTO150
·90 POKE49456,1:INPUT"DISK OR TAPE (D/T)";C$:IFC$<>"D"THE
ND=1:D$="TAPE"
·100 POKE49457,D:D$=D$+" DRIVE":PRINT:INPUT"FILE NUMBER (
0-99)";N
·110 N$=RIGHT$(STR$(N),2):IFN<10THENN$=CHR$(48)+CHR$(N+48
)
·120 F$="BACKUP":PRINT:INPUT"FILENAME";F$:F$=N$+LEFT$(F$,
14):L=LEN(F$)
·130 POKE49458,L:FORJ=1TOL:POKE49458+J,ASC(MID$(F$,J,1)):
NEXTJ:PRINT
·140 PRINT"SAVING DEVICE ** "D$:PRINT"STARTING WITH ** "F
$
·150 PRINT:INPUT"IS THIS CORRECT (Y/N)";B$:IFB$<>"Y"THEN6
0
·160 POKE770,131:POKE771,164:SYS49152:END
·170 DATA169,79,32,210,255,162,38,160,192,204,3,3,1507
·180 DATA208,10,162,131,160,164,169,70,32,210,255,44,1615
·190 DATA169,78,32,210,255,142,2,3,140,3,3,76,1113
·200 DATA36,193,32,96,165,134,122,132,123,32,115,0,1180
·210 DATA170,240,243,162,255,134,58,144,3,76,150,164,1799
·220 DATA32,107,169,32,121,165,173,0,2,240,5,169,1215
·230 DATA79,141,2,3,76,162,164,169,0,133,2,133,1064
·240 DATA251,133,252,133,254,24,101,20,69,254,230,254,197
5
·250 DATA24,101,21,69,254,170,230,254,164,252,185,0,1724
·260 DATA2,133,253,201,34,208,6,165,2,73,255,133,1465
·270 DATA2,201,32,208,4,165,2,240,8,138,24,101,1125
·280 DATA253,69,254,170,44,198,254,230,252,164,253,208,23
49
·290 DATA213,138,41,240,74,74,74,74,24,105,129,141,1327
·300 DATA44,193,138,41,15,24,105,129,141,45,193,162,1230
·310 DATA0,189,43,193,240,12,157,0,4,173,134,2,1147
·320 DATA157,0,216,232,208,239,169,38,141,2,3,173,1578
·330 DATA48,193,240,23,165,161,201,212,176,4,165,160,1748
·340 DATA240,13,238,32,208,160,0,32,225,255,208,6,1617
·350 DATA32,33,193,76,38,192,232,208,242,200,208,239,1893
·360 DATA32,68,229,169,0,168,174,49,193,32,186,255,1555
·370 DATA173,50,193,162,51,160,193,32,189,255,169,43,1670
·380 DATA166,45,164,46,32,216,255,162,1,189,51,193,1520
·390 DATA168,200,152,201,58,144,2,169,48,157,51,193,1543
·400 DATA201,48,208,3,202,16,234,32,33,193,76,116,1362
·410 DATA164,206,32,208,169,0,170,168,76,219,255,160,1827
·420 DATA1,1,160,0,0,65,72,79,89,33,0,0,500
```

C-128 BUG REPELLENT

```
·10 PRINTCHR$(147)"LOADING AND CHECKING THE DATA[3"."]":J
=4864
·20 FORB=0TO11:READA:IFA<0ORA>255THEN40
·30 POKEJ+B,A:X=X+A:NEXTB:READA:IFA=XTHEN50
·40 PRINT:PRINT"ERROR IN DATA LINE:"PEEK(66)*256+PEEK(65)
:END
·50 X=0:J=J+12:IFJ<5213THEN20
·60 POKE208,0:POKE5213,0:A$="Y":B$=A$:C$="D":D$="DISK":D=
8:PRINTCHR$(147)
·70 INPUT"DO YOU WANT AUTOMATIC SAVES (Y/N)";A$:PRINT:IFA
$="Y"THEN90
·80 PRINT"NO AUTOMATIC SAVES[3"."]":GOTO150
·90 POKE5213,1:INPUT"DISK OR TAPE (D/T)";C$:IFC$<>"D"THEN
D=1:D$="TAPE"
·100 POKE5214,D:D$=D$+" DRIVE":PRINT:INPUT"FILE NUMBER (0
-99)";N
·110 N$=RIGHT$(STR$(N),2):IFN<10THENN$=CHR$(48)+CHR$(N+48
)
·120 F$="BACKUP":PRINT:INPUT"FILENAME";F$:F$=N$+LEFT$(F$,
14):L=LEN(F$)
·130 POKE5215,L:FORJ=1TOL:POKE5215+J,ASC(MID$(F$,J,1)):NE
XTJ:PRINT
·140 PRINT"SAVING DEVICE ** "D$:PRINT"STARTING WITH ** "F
$
·150 PRINT:INPUT"IS THIS CORRECT (Y/N)";B$:IFB$<>"Y"THEN6
0
·160 POKE770,198:POKE771,77:SYS4864:END
·170 DATA32,58,20,169,41,162,19,236,3,3,208,4,955
·180 DATA169,198,162,77,141,2,3,142,3,3,224,19,1143
·190 DATA208,7,32,125,255,79,78,0,96,32,125,255,1292
·200 DATA79,70,70,0,96,162,0,134,251,189,0,2,1053
·210 DATA240,19,201,48,144,9,201,58,176,5,133,251,1485
·220 DATA232,208,238,134,252,165,251,208,3,76,198,77,2042
·230 DATA169,0,166,235,164,236,133,253,133,254,142,47,193
2
·240 DATA20,140,48,20,24,101,22,69,254,230,254,24,1206
·250 DATA101,23,69,254,170,230,254,164,252,185,0,2,1704
·260 DATA133,251,201,34,208,6,165,253,73,255,133,253,1965
·270 DATA201,32,208,4,165,253,240,8,138,24,101,251,1625
·280 DATA69,254,170,44,198,254,230,252,164,251,208,213,23
07
·290 DATA138,41,240,74,74,74,74,24,105,65,141,88,1138
·300 DATA20,138,41,15,24,105,65,141,89,20,32,79,769
·310 DATA20,189,85,20,240,6,32,210,255,232,208,245,1742
·320 DATA174,47,20,172,48,20,24,32,240,255,173,93,1298
·330 DATA20,240,27,165,161,201,212,176,4,165,160,240,1771
·340 DATA17,32,65,20,238,32,208,238,1,214,32,225,1322
·350 DATA255,208,6,32,49,20,76,198,77,232,208,242,1603
·360 DATA200,208,239,32,66,193,173,95,20,162,96,160,1644
·370 DATA20,32,189,255,169,0,170,32,104,255,169,0,1395
·380 DATA174,94,20,168,32,186,255,169,45,174,16,18,1351
·390 DATA172,17,18,32,216,255,162,1,189,96,20,168,1346
·400 DATA200,152,201,58,144,2,169,48,157,96,20,201,1448
·410 DATA48,208,3,202,16,234,32,49,20,141,0,2,955
·420 DATA76,183,77,58,59,32,65,20,206,32,208,206,1222
·430 DATA1,214,169,0,170,168,76,219,255,32,79,20,1403
·440 DATA169,26,141,0,214,173,0,214,16,251,96,162,1462
·450 DATA0,142,0,255,96,19,18,32,32,32,32,146,804
·460 DATA0,1,0,0,65,72,79,89,33,0,0,0,339
```

FLANKSPEED FOR THE C-64 By GORDON F. WHEAT

Flankspeed will allow you to enter machine language *Ahoy!* programs without any mistakes. Once you have typed the program in, save it for future use. While entering an ML program with *Flankspeed* there is no need to enter spaces or hit the carriage return. This is all done automatically. If you make an error in a line a bell will ring and you will be asked to enter it again. To LOAD in a program Saved with *Flankspeed* use LOAD "name",1,1 for tape, or LOAD "name",8,1 for disk. The function keys may be used after the starting and ending addresses have been entered.

f1–SAVEs what you have entered so far.

f3–LOADs in a program worked on previously.

f5–To continue on a line you stopped on after LOADing in the previous saved work.

f7–Scans through the program to locate a particular line, or to find out where you stopped the last time you entered the program. It temporarily freezes the output as well.

```
·100 POKE53280,12:POKE53281,11                              OP
·105 PRINT"[CLEAR][c 8][RVSON][15" "]FLANKSPEED[15" "]";    FP
·110 PRINT"[RVSON][5" "]MISTAKEPROOF ML ENTRY PROGRAM[6" "
 ]"                                                         JP
·115 PRINT"[RVSON][9" "]CREATED BY G. F. WHEAT[9" "]"       FA
·120 PRINT"[RVSON][3" "]COPR. 1987, ION INTERNATIONAL INC.
 [3" "]"                                                    AJ
·125 FORA=54272TO54296:POKEA,0:NEXT                         ND
·130 POKE54272,4:POKE54273,48:POKE54277,0:POKE54278,249:PO
 KE54296,15                                                 NP
·135 FORA=680TO699:READB:POKEA,B:NEXT                       FL
·140 DATA169,251,166,253,164,254,32,216,255,96              FF
·145 DATA169,0,166,251,164,252,32,213,255,96                EK
·150 B$="STARTING ADDRESS IN HEX":GOSUB430:AD=B:SR=B        KP
·155 GOSUB480:IFB=0THEN150                                  OE
·160 POKE251,T(4)+T(3)*16:POKE252,T(2)+T(1)*16              AM
·165 B$="ENDING ADDRESS IN HEX":GOSUB430:EN=B               PE
·170 GOSUB470:IFB=0THEN150                                  PG
·175 POKE254,T(2)+T(1)*16:B=T(4)+1+T(3)*16                  GM
·180 IFB>255THENB=B-255:POKE254,PEEK(254)+1                 HG
·185 POKE253,B:PRINT                                        EC
·190 REM  GET HEX LINE                                      ED
·195 GOSUB495:PRINT": [c P][LEFT]";:FORA=0TO8               KD
·200 FORB=0TO1:GOTO250                                      IH
·205 NEXTB                                                  IJ
·210 A%(A)=T(1)+T(0)*16:IFAD+A-1=ENTHEN340                  FA
·215 PRINT" [c P][LEFT]";                                   EG
·220 NEXTA:T=AD-(INT(AD/256)*256):PRINT" "                  II
·225 FORA=0TO7:T=T+A%(A):IFT>255THENT=T-255                 GL
·230 NEXT                                                   GI
·235 IFA%(8)<>TTHENGOSUB375:GOTO195                         FL
·240 FORA=0TO7:POKEAD+A,A%(A):NEXT:AD=AD+8:GOTO195          IM
·245 REM     GET HEX INPUT                                  PA
·250 GETA$:IFA$=""THEN250                                   GA
·255 IFA$=CHR$(20)THEN305                                   GO
·260 IFA$=CHR$(133)THEN535                                  LM
·265 IFA$=CHR$(134)THEN560                                  IG
·270 IFA$=CHR$(135)THENPRINT" ":GOTO620                     HO
·275 IFA$=CHR$(136)THENPRINT" ":GOTO635                     HE
·280 IFA$>"@"ANDA$<"G"THENT(B)=ASC(A$)-55:GOTO295           MI
·285 IFA$>"/"ANDA$<":"THENT(B)=ASC(A$)-48:GOTO295           DJ
·290 GOSUB415:GOTO250                                       JA
·295 PRINTA$"[c P][LEFT]";                                  PK
·300 GOTO205                                                FA
·305 IFA>0THEN320                                           BI
·310 A=-1:IFB=1THEN330                                      BB
·315 GOTO220                                                FA
·320 IFB=0THENPRINTCHR$(20);CHR$(20);:A=A-1                 BF
·325 A=A-1                                                  FK
·330 PRINTCHR$(20);:GOTO220                                 PH
·335 REM LAST LINE                                          CP
·340 PRINT" ":T=AD-(INT(AD/256)*256)                        KH
·345 FORB=0TOA-1:T=T+A%(B):IFT>255THENT=T-255               OD
·350 NEXT                                                   OB
·355 IFA%(A)<>TTHENGOSUB375:GOTO195                         LH
·360 FORB=0TOA-1:POKEAD+B,A%(B):NEXT                        BO
·365 PRINT:PRINT"YOU ARE FINISHED!":GOTO535                 MB
·370 REM BELL AND ERROR MESSAGES                            LM
·375 PRINT:PRINT"LINE ENTERED INCORRECTLY":PRINT:GOTO415    JK
·380 PRINT:PRINT"INPUT A 4 DIGIT HEX VALUE!":GOTO415        PG
·385 PRINT:PRINT"ENDING IS LESS THAN STARTING!":B=0:GOTO41
 5                                                          IK
·390 PRINT:PRINT"ADDRESS NOT WITHIN SPECIFIED RANGE!":B=0:
 GOTO415                                                    HK
·395 PRINT:PRINT"NOT ZERO PAGE OR ROM!":B=0:GOTO415         DM
·400 PRINT"?ERROR IN SAVE":GOTO415                          JK
·405 PRINT"?ERROR IN LOAD":GOTO415                          IO
·410 PRINT:PRINT:PRINT"END OF ML AREA":PRINT                JO
·415 POKE54276,17:POKE54276,16:RETURN                       BF
·420 OPEN15,8,15:INPUT#15,A,A$:CLOSE15:PRINTA$:RETURN       DH
·425 REM GET FOUR DIGIT HEX                                 IM
·430 PRINT:PRINTB$;:INPUTT$                                 OL
·435 IFLEN(T$)<>4THENGOSUB380:GOTO430                       JD
·440 FORA=1TO4:A$=MID$(T$,A,1):GOSUB450:IFT(A)=16THENGOSUB
 380:GOTO430                                                AK
·445 NEXT:B=(T(1)*4096)+(T(2)*256)+(T(3)*16)+T(4):RETURN    KB
·450 IFA$>"@"ANDA$<"G"THENT(A)=ASC(A$)-55:RETURN            GM
·455 IFA$>"/"ANDA$<":"THENT(A)=ASC(A$)-48:RETURN            NJ
·460 T(A)=16:RETURN                                         IC
·465 REM ADDRESS CHECK                                      OL
·470 IFAD>ENTHEN385                                         HO
·475 IFB<SRORB>ENTHEN390                                    LE
·480 IFB<256OR(B>40960ANDB<49152)ORB>53247THEN395           OB
·485 RETURN                                                 HE
·490 REM ADDRESS TO HEX                                     PM
·495 AC=AD:A=4096:GOSUB520                                  AP
·500 A=256:GOSUB520                                         NF
·505 A=16:GOSUB520                                          LG
·510 A=1:GOSUB520                                           HE
·515 RETURN                                                 JD
·520 T=INT(AC/A):IFT>9THENA$=CHR$(T+55):GOTO530             OC
·525 A$=CHR$(T+48)                                          JI
·530 PRINTA$;:AC=AC-A*T:RETURN                              AA
·535 A$="**SAVE**":GOSUB585                                 IC
·540 OPEN1,T,1,A$:SYS680:CLOSE1                             AB
·545 IFST=0THENEND                                          FB
·550 GOSUB400:IFT=8THENGOSUB420                             PM
·555 GOTO535                                                FI
·560 A$="**LOAD**":GOSUB585                                 PE
·565 OPEN1,T,0,A$:SYS690:CLOSE1                             PO
·570 IFST=64THEN195                                         OI
·575 GOSUB405:IFT=8THENGOSUB420                             CO
·580 GOTO560                                                GN
·585 PRINT" ":PRINTTAB(14)A$                                KA
·590 PRINT:A$="":INPUT"FILENAME";A$                         IO
·595 IFA$=""THEN590                                         HK
·600 PRINT:PRINT"TAPE OR DISK?":PRINT                       HL
·605 GETB$:T=1:IFB$="D"THENT=8:A$="@0:"+A$:RETURN           NP
·610 IFB$<>"T"THEN605                                       KO
·615 RETURN                                                 PH
·620 B$="CONTINUE FROM ADDRESS":GOSUB430:AD=B               DD
·625 GOSUB475:IFB=0THEN620                                  NK
·630 PRINT:GOTO195                                          HN
·635 B$="BEGIN SCAN AT ADDRESS":GOSUB430:AD=B               FK
·640 GOSUB475:IFB=0THEN635                                  LN
·645 PRINT:GOTO670                                          HI
·650 FORB=0TO7:AC=PEEK(AD+B):GOSUB505:IFAD+B=ENTHENAD=SR:G
 OSUB410:GOTO195                                            LM
·655 PRINT" ";:NEXTB                                        LE
·660 PRINT:AD=AD+8                                          CD
·665 GETB$:IFB$=CHR$(136)THEN195                            JD
·670 GOSUB495:PRINT": ";:GOTO650                            KE
```

IMPORTANT! Letters on white background are **Bug Repellent** line codes. **Do not enter them!** Pages 77 and 78 explain these codes and provide other essential information on entering ***Ahoy!*** programs. Refer to these pages **before** entering any programs!

BATTER UP!
FROM PAGE 31

```
•5 REM    BATTER UP! BY JC HILTY 3/87          BE
•10 GOSUB745                                    DE
•15 GOSUB515                                    CM
•20 B=0:K=0:OT=0:R=0:X=1                        FC
•25 IFIN=10THEN1050                             LC
•30 POKE2040,200:POKE2042,202:FORT=0TO500
 :NEXT                                          PO
•35 PRINT"[HOME][DOWN][RVSON][BLACK]"TAB(
 2)B;TAB(5)K;TAB(9)OT;TAB(14)IN                 FH
•40 IFOT=3THEN420                               JI
•45 IFK=3THENOT=OT+1:GOSUB105:K=0:B=0:GOT
 035                                            GL
•50 IFB=4THENGOSUB380:B=0:K=0:GOSUB105:GO
 TO35                                           HF
•55 PRINT"[HOME][7"[DOWN]"]"TAB(22)"[12"
 "]"                                            IA
•60 PRINT"[HOME][7"[DOWN]"][YELLOW]"TAB(2
 2)P$(X);" ";PA(X):GOSUB690:FORT=0TO1000:
 NEXT                                           ND
•65 POKE2042,203:FORT=0TO300:NEXT:POKE204
 2,204:FORT=0TO200:NEXT                         IF
•70 Z=INT(8*RND(1)+1)                           FO
•75 ONZGOSUB120,130,120,140,120,150,120,1
 60                                             PH
•80 W=206                                       PH
•85 FB=-((PEEK(56320)AND16)=0)                  LH
•90 IFFB=1THENPOKE2040,208:POKEV+21,31:GO
 TO170                                          AB
•95 POKE2044,W:FORT=0TO50:NEXT:W=W+1:IFW=
 208THEN185                                     HJ
•100 GOTO85                                     FN
•105 X=X+1:IFX=10THENX=1:RETURN                 IA
•110 RETURN                                     PM
•115 REM    COMPUTER PITCHES                    FK
•120 POKEV+8,88:POKEV+9,162:POKEV+21,23:F
 ORT=0TO50:NEXT:POKEV+9,172                     MK
•125 RETURN                                     AL
•130 POKEV+8,88:POKEV+9,162:POKEV+21,23:F
 ORT=0TO50:POKEV+9,145                          GD
•135 RETURN                                     BF
•140 POKEV+8,88:POKEV+9,162:POKEV+21,23:F
 ORT=0TO50:POKEV+9,172:POKEV+8,120              JK
•145 RETURN                                     BP
•150 POKEV+8,88:POKEV+9,162:POKEV+21,23:F
 ORT=0TO50:POKEV+9,200:POKEV+8,70               IL
•155 RETURN                                     CJ
•160 POKEV+8,88:POKEV+9,162:POKEV+21,23:F
 ORT=0TO50:POKEV+9,200:POKEV+8,100              AC
•165 RETURN                                     DD
•170 FORT=0TO150:NEXT:IFZ=1ORZ=3ORZ=5ORZ=
 7THEN205                                       EP
•175 K=K+1:FORT=0TO200:NEXT:POKEV+21,7:E$
 ="STRIKE"                                      EI
•180 GOTO260                                    NE
•185 IFZ=1ORZ=3ORZ=5ORZ=7THENE$="STRIKE":
 K=K+1:FORT=0TO500:NEXT:GOTO195                 AL
•190 E$="BALL":B=B+1:FORT=0TO500:NEXT           IN
•195 POKEV+21,7:GOTO260                         HC
•200 REM    BATTER HITS BALL                    FN
•205 GOSUB705:POKEV+21,7:E=RND(5)+PA(X)/1
 000                                            PD
•210 IFE>1.1THENE$="HOME RUN":GOSUB285:B=
 0:K=0:GOSUB105:GOTO260                         NF
•215 IFE>.9THENE$="TRIPLE":GOSUB310:B=0:K
 =0:GOSUB105:GOTO260                            OC
•220 IFE>.8THENE$="DOUBLE":GOSUB335:B=0:K
 =0:GOSUB105:GOTO260                            PA
•225 IFE>.6THENE$="SINGLE":GOSUB380:B=0:K
 =0:GOSUB105:GOTO260                            JO
•230 IFE>.5THENE$="FOUL BALL":GOSUB245:GO
 TO260                                          OG
•235 IFE>.3THENE$="GROUND OUT":OT=OT+1:GO
 SUB105:B=0:K=0:GOTO260                         LJ
•240 E$="FLY OUT":OT=OT+1:B=0:K=0:GOSUB10
 5:GOTO260                                      BA
•245 IFK<2THENK=K+1:RETURN                      DL
•250 RETURN                                     II
•255 REM    DISPLAY HITTING RESULTS             KN
•260 PRINT"[HOME][13"[DOWN]"][YELLOW]"TAB
 (27)E$                                         CL
•265 FORT=0TO2500:NEXT                          ND
•270 PRINT"[HOME][13"[DOWN]"][RVSON][GREE
 N]"TAB(27)"[10" "]"                            CE
•275 GOTO25                                     AD
•280 REM    HOMERUN                             MB
•285 IFR=0THENYS=YS+1:R=0:GOSUB455:GOSUB5
 10:RETURN                                      IK
•290 IFR=1ORR=2ORR=4THENYS=YS+2:R=0:GOSUB
 455:GOSUB510:RETURN                            GJ
•295 IFR=3ORR=5ORR=6THENYS=YS+3:R=0:GOSUB
 455:GOSUB510:RETURN                            PI
•300 IFR=7THENYS=YS+4:R=0:E$="GRAND SLAM"
 :GOSUB455:GOSUB510:RETURN                      BF
•305 REM    TRIPLE                              JK
•310 IFR=0THENR=4:GOSUB455:RETURN               PD
•315 IFR=1ORR=2ORR=4THENYS=YS+1:R=4:GOSUB
 455:GOSUB510:RETURN                            CJ
•320 IFR=3ORR=5ORR=6THENYS=YS+2:R=4:GOSUB
 455:GOSUB510:RETURN                            IA
•325 IFR=7THENYS=YS+3:R=4:GOSUB455:GOSUB5
 10:RETURN                                      CN
•330 REM    DOUBLE                              JK
•335 IFR=0THENR=2:GOSUB455:RETURN               BM
•340 IFR=1THENR=6:GOSUB455:RETURN               CK
•345 IFR=2THENYS=YS+1:R=2:GOSUB455:GOSUB5
 10:RETURN                                      KK
•350 IFR=3THENYS=YS+1:R=6:GOSUB455:RETURN       IF
•355 IFR=4THENYS=YS+1:R=2:GOSUB455:GOSUB5
```

```
10:RETURN                                      NG
•360 IFR=5THENYS=YS+1:R=6:GOSUB455:GOSUB5
10:RETURN                                      PI
•365 IFR=6THENYS=YS+2:R=2:GOSUB455:GOSUB5
10:RETURN                                      PB
•370 IFR=7THENYS=YS+2:R=6:GOSUB455:GOSUB5
10:RETURN                                      CD
•375 REM    SINGLE OR WALK                     JP
•380 IFR=0THENR=1:GOSUB455:RETURN              AA
•385 IFR=1THENR=3:GOSUB455:RETURN              FE
•390 IFR=3THENR=7:GOSUB455:RETURN              JF
•395 IFR=4THENYS=YS+1:R=1:GOSUB455:GOSUB5
10:RETURN                                      EP
•400 IFR=5THENYS=YS+1:R=3:GOSUB455:GOSUB5
10:RETURN                                      DL
•405 IFR=6THENYS=YS+1:R=5:GOSUB455:GOSUB5
10:RETURN                                      NP
•410 IFR=7THENYS=YS+1:R=7:GOSUB455:GOSUB5
10:RETURN                                      HD
•415 IFR=2THENR=5:GOSUB455:RETURN              CP
•420 I=INT(4*RND(6))                           PJ
•425 PRINT"[HOME][13"[DOWN]"][YELLOW]"TAB
(22)"COMPUTER "I:GOSUB720                      EO
•430 CS=CS+I:PRINT"[HOME][13"[DOWN]"][RVS
ON][GREEN]"TAB(22)"[12" "]"                    GL
•435 PRINT"[HOME][4"[DOWN]"][YELLOW]"TAB(
32)CS:GOSUB105                                 BB
•440 IN=IN+1:B=0:K=0:R=0:OT=0:GOSUB455         HN
•445 GOTO25                                    LF
•450 REM    PLACE BASERUNNERS                  AP
•455 R=R+1:ONRGOSUB465,470,475,480,485,49
0,495,500                                      DP
•460 R=R-1:RETURN                              LI
•465 POKEFI,160:POKESN,160:POKETH,160:RET
URN                                            BK
•470 POKEFI,86:POKESN,160:POKETH,160:RETU
RN                                             BL
•475 POKESN,86:POKEFI,160:POKETH,160:RETU
RN                                             AM
•480 POKETH,160:POKESN,86:POKEFI,86:RETUR
N                                              LG
•485 POKETH,86:POKESN,160:POKEFI,160:RETU
RN                                             OG
•490 POKESN,160:POKETH,86:POKEFI,86:RETUR
N                                              LI
•495 POKEFI,160:POKESN,86:POKETH,86:RETUR
N                                              JB
•500 POKEFI,86:POKESN,86:POKEFI,86:RETURN FJ
•505 REM    UPDATE SCORE                       KN
•510 PRINT"[HOME][3"[DOWN]"][YELLOW]"TAB(
32)YS:RETURN                                   GI
•515 REM    PLAYING FIELD                      NE
•520 POKE53281,8:POKE53280,0                   BP
•525 PRINT"[CLEAR][RVSON][BLACK][3" "]B
S  OUT  INN[3" "][c 7][19" "]"                 DD
•530 PRINT"[RVSON][BLACK][20" "][c 7][19"
 "]"                                           ML
•535 PRINT"[RVSON][BLUE][20" "][c 7] [BLA
CK][16" "][c 7]  "                             ME
•540 FORT=0TO2:PRINT"[RVSON][BLUE][20" "]
[c 7] [BLACK] "TAB(36)" [c 7]  ":NEXT          BJ
•545 PRINT"[RVSON][c 2][20" "][c 7] [BLAC
K] "TAB(36)" [c 7]  "                          LL
•550 FORT=0TO1:PRINT"[RVSON][GREEN][20" "
][c 7] [BLACK] "TAB(36)" [c 7]  ":NEXT         KH
•555 PRINT"[RVSON][GREEN][20" "][c 7] [BL
ACK][16" "][c 7]  "                            MJ
•560 PRINTTAB(20)"[RVSON][c 7]  [BLACK] [
c 7][12" "][BLACK] [c 7][3" "]"                GE
•565 PRINT"[RVSON][GREEN][20" "][BLUE]  [
BLACK] [BLUE][12" "][BLACK] [BLUE][3" "]
"                                              EB
•570 PRINT"[RVSON][GREEN][21" "][s N][15"
 "][s M] "                                     FD
•575 PRINT"[RVSON][6" "][RVSOFF][sEP][6"
"][c *][RVSON][6" "][s N][17" "][s M]"         AL
•580 PRINT"[RVSON][20" "][s M][17" "][s N
]"                                             OD
•585 PRINT"[c *][RVSON][18" "][RVSOFF][sE
P][RVSON] [s M][15" "][s N] "                  AD
•590 PRINT" [c *][RVSON][16" "][RVSOFF][s
EP] [RVSON]  [s M][13" "][s N]  "              KE
•595 PRINT"  [c *][RVSON][14" "][RVSOFF][
sEP]  [RVSON][3" "][s M][5" "][WHITE] [G
REEN][5" "][s N][3" "]"                        AP
•600 PRINT"[3" "][WHITE][5"[c P]"][4" "][
5"[c P]"][3" "][RVSON][GREEN][4" "][s M]
[3" "][s N] [s M][3" "][s N][4" "]"            JG
•605 PRINT"  [WHITE][s N][4" "][s N][4" "
][s M][4" "][s M]  [RVSON][GREEN][5" "][
s M] [s N][3" "][s M] [s N][5" "]"             HB
•610 PRINT" [WHITE][s N][4" "][s N] [RVSO
N][4" "][RVSOFF] [s M][4" "][s M] [RVSON
][GREEN][6" "][WHITE] [GREEN]  [s Z]  [W
HITE] [GREEN][6" "]"                           DL
•615 PRINT"[WHITE][s N][4" "][s N]  [c *]
[RVSON]  [RVSOFF][sEP]  [s M][4" "][s M]
[RVSON][GREEN][7" "][s M][3" "][s N][7"
"]"                                            FL
•620 PRINT"[WHITE][4" "][s N]"TAB(15)"[s
M][4" "][RVSON][GREEN][8" "][s M] [s N][
8" "]"                                         FM
•625 PRINT"[WHITE][4"[c Y]"]"TAB(16)"[4"[
c Y]"][RVSON][GREEN][9" "][WHITE] [GREEN
][9" "]"                                       FP
•630 FORX=2004TO2022:POKEX,160:POKEX+5427
2,5:NEXT                                       KO
•635 X=1063:FORT=0TO24:POKEX,160:POKEX+54
272,0:X=X+40:NEXT                              BD
•640 PRINT"[HOME][3"[DOWN]"][YELLOW]"TAB(
22)T$;TAB(32)YS:PRINT"[HOME][4"[DOWN]"]"
TAB(22)C$;TAB(32)CS                            CH
•645 PRINT"[HOME][6"[DOWN]"]"TAB(22)"NOW
BATTING"                                       JF
•650 POKEV+16,0:POKEV+27,0:POKEV+23,15:PO
KEV+29,15:POKEV+28,7                           OC
```

```
·655 POKEV+37,10:POKEV+38,0                    OA
·660 POKE2040,200:POKEV+39,7:POKEV,30:POK
 EV+1,150                                      HM
·665 POKE2041,201:POKEV+40,1:POKEV+2,30:P
 OKEV+3,192                                    NA
·670 POKE2042,202:POKEV+41,6:POKEV+4,78:P
 OKEV+5,120                                    LA
·675 POKE2043,209:POKEV+42,0:POKEV+6,64:P
 OKEV+7,160                                    LP
·680 POKE2044,205:POKEV+43,1:POKEV+8,80:P
 OKEV+9,192                                    MI
·685 POKEV+21,7:RETURN                         MA
·690 FORT=0TO24:POKES+T,0:NEXT                 PG
·695 POKES,150:POKES+1,200:POKES+5,8:POKE
 S+6,248:POKES+24,15:POKES+4,17                AM
·700 FORT=0TO0500:NEXT:POKES+4,16:RETURN       EP
·705 FORT=0TO24:POKES+T,0:NEXT                 MH
·710 POKES+24,15:POKES+1,80:POKES+5,80:PO
 KES+6,245                                     MN
·715 POKES+4,129:FORT=0TO100:NEXT:POKES+4
 ,128:RETURN                                   CK
·720 FORT=0TO24:POKES+T,0:NEXT                 MA
·725 POKES+18,128:POKES,75:POKES+6,240:PO
 KES+14,12:POKES+15,250:POKES+24,207           NH
·730 FORL=0TO30:POKES+4,17:POKES+1,PEEK(S
 +27)                                          HJ
·735 FORT=0TO100:NEXT:NEXT:POKES+4,0           ME
·740 RETURN                                    HE
·745 POKE53281,12:POKE53280,0:PRINT"[CLEA
 R][6"[DOWN]"][BLUE]"TAB(14)"[12"*"]"          EG
·750 PRINTTAB(14)"*[10" "]*":PRINTTAB(14)
 "*BATTER UP!*"                                NL
·755 PRINTTAB(14)"*[10" "]*":PRINTTAB(14)
 "[12"*"]":PRINTTAB(19)"BY"                    GN
·760 PRINTTAB(16)"JC HILTY":PRINT:PRINT        PE
·765 PRINT"PLEASE WAIT WHILE WE BUY SOME
 PEANUTS.."                                    HC
·770 DIMPA(9),P$(9):YS=0:CS=0:IN=1:SN=173
 3:TH=1850:FI=1856                             FB
·775 S=54272:V=53248                           DO
·780 FORX=12800TO13438:READA:POKEX,A:NEXT JB
·785 FORX=1TO9:READP$(X):NEXT                  CE
·790 PRINT"[CLEAR][DOWN]PLEASE ENTER THE
 NAME OF YOUR TEAM[4"."]"                      CA
·795 PRINT"[4"."]UP TO 10 LETTERS."            OK
·800 INPUTT$:IFLEN(T$)>10THEN790               JM
·805 IFLEN(T$)<1THENT$="PITTSBURGH"            DD
·810 PRINT"[CLEAR][DOWN]PLEASE ENTER THE
 COMPUTER TEAM'S NAME.."                       HH
·815 PRINT"..UP TO 10 LETTERS."                JF
·820 INPUTC$:IFLEN(C$)>10THEN810               JF
·825 IFLEN(C$)<1THENC$="NEW YORK"              IH
·830 FORX=1TO9:PA(X)=INT(300*RND(5)+100):
 NEXT                                          NJ
·835 PRINT"[CLEAR][DOWN]LINEUPS AND BATTI
 NG AVERAGES":PRINT:PRINTT$:PRINT              AE
·840 FORX=1TO9:PRINTP$(X);" ";PA(X):NEXT:
 PRINT:PRINT                                   CC
·845 PRINT"PRESS P TO PRINT LINEUP ON PRI
 NTER[4"."]"                                   EA
·850 PRINT"[4"."]ANY OTHER KEY TO CONTINU
 E."                                           HP
·855 GETA$:IFA$=""THEN855                      NJ
·860 IFA$="P"THENGOSUB1030                     KB
·865 RETURN                                    PC
·870 REM    SPRITE DATA                        NE
·875 DATA 240,42,0,240,42,0,240,42,128,25
 2,42,160,60,41,64,60,41,64,63,53,64           IO
·880 DATA 3,53,64,3,205,0,2,234,0,10,250,
 128,10,186,128,14,190,128,53,94,128           DJ
·885 DATA 53,93,240,61,171,112,10,169,112
 ,10,170,240,15,170,0,15,252,0,15,252,0        OJ
·890 DATA 0                                    CK
·895 DATA 10,170,0,10,170,0,10,170,128,10
 ,171,128,10,174,128,2,174,128                 PD
·900 DATA 2,174,128,2,174,128,2,174,128,2
 ,174,0,2,174,0,10,170,0,10,186,0              FB
·905 DATA 10,191,240,10,188,0,10,128,0,11
 ,192,0,15,252,0,15,0,0,0,0,0,0,0,0            ML
·910 DATA 0,10,0,0,42,0,0,5,0,0,5,0,0,1,0
 ,0,10,128,0,42,160,0,170,168,0,138,136        LO
·915 DATA 0,138,136,0,169,168,0,15,192,0,
 10,128,0,10,128,0,8,128,0,8,128               FN
·920 DATA 0,8,128,0,12,192,0,12,192,0,60,
 240,0,0,0,0                                   HC
·925 DATA 0,10,0,0,42,0,0,5,0,0,5,0,0,1,0
 ,0,10,0,0,10,0,0,10,0,0,10,0,0,10,0           BN
·930 DATA 0,10,0,0,15,0,0,1,0,0,2,0,0,2,0
 ,0,2,0,0,2,0,0,3,0,0,3,0,0,15,0               JF
·935 DATA 0,0,0,0                              HI
·940 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
 ,0,0,0,10,0,0,10,36,0,165,128,0,170,0         NO
·945 DATA 0,138,48,0,138,128,0,138,0,0,66
 ,0,0,2,0,0,2,0,0,3,0,0,0,0,0,0,0              AC
·950 DATA 0,0,0,0,0,0,0                        IC
·955 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
 ,0,0,0,0,0,0,24,0,0,60,0,0,24,0               JE
·960 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
 ,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0            JH
·965 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
 ,0,0,0,24,0,0,60,0,0,126,0,0,60,0             KG
·970 DATA 0,24,0,0,0,0,0,0,0,0,0,0,0,0,0,
 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0           BI
·975 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
 ,0,0,0,60,0,0,255,0,0,255,0,0,255,0           JC
·980 DATA 0,255,0,0,60,0,0,0,0,0,0,0,0,0,
 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0           PP
·985 DATA 0,42,0,0,42,0,0,42,128,0,42,160
 ,0,41,64,0,41,64,0,21,64,0,21,64              AM
·990 DATA 0,165,0,2,170,0,2,170,128,2,170
 ,128,2,170,128,2,170,128,10,170,128           CN
·995 DATA 10,170,128,10,170,128,10,170,0,
 15,234,0,15,252,0,15,252,0,0                  AA
·1000 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
 0,0,0,0,0,0,0,0,0,0,0,124,135,255,252         MG
·1005 DATA 255,255,252,128,63,252,0,0,0,0
```

```
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0      EC
•1010 DATA 0,0,0,0,0,0,0,0                  NO
•1015 REM    PLAYER NAMES                   KL
•1020 DATA ROCKY,SPIKE,PETE,WILLIE,HAMMER
,JIM,MOOSE,BOOMER,HAWK                      EG
•1025 REM    PRINTOUT LINEUP                BN
•1030 PRINT"[CLEAR][DOWN][DOWN]MAKE SURE
 PRINTER IS ON[4"."]":PRINT"[4"."]PRESS A
 NY KEY."                                   DG
•1035 GETA$:IFA$=""THEN1035                 MC
•1040 OPEN3,4,0:PRINT#3,"LINEUP AND AVERA
 GES FOR ";T$                               MA
•1045 FORX=1TO9:PRINT#3,P$(X);" ";PA(X):N
 EXT:CLOSE3                                 JF
•1046 PRINT:PRINT"PRESS ANY KEY TO BEGIN
 GAME."                                     AO
•1047 GETA$:IFA$=""THEN1047                 LB
•1048 RETURN                                KK
•1049 REM    GAME OVER                      AK
•1050 POKEV+21,0                            HB
•1055 POKE53281,0:POKE53280,0:PRINT"[CLEA
 R][8"[DOWN]"][c 7]"TAB(14)"[12"*"]"        EN
•1060 PRINTTAB(14)"*SCOREBOARD*":PRINTTAB
 (14)"[12"*"]":PRINT:PRINT                  DJ
•1065 PRINTTAB(14)T$;YS:PRINTTAB(14)C$;CS   HM
•1070 END                                   LC
```

GREEN SCREEN BLUES

FROM PAGE 32

SCRNSWAP.ASM

```
1 "************************************
2 "*       SCRNSWAP.ASM
3 "*     RUPERT REPORT #44
4 "************************************
5 REM  USE INTERRUPT ROUTINE TO READ
6 REM  40/80 COLUMN MODE OF C-128
7 REM  AND SWITCH A RELAY TO ENABLE
8 REM  THE SELECTED VIDEO OUTPUT
10 "************************************
11 "************************************
105 "SYS EQU $0A00        ;SYSTEM VECTOR
106 "IRQ EQU $0314        ;IRQ VECTOR
107 "MODEFLG EQU $D7
108 "*      BIT 7 = 1/0 = 80/40 COL MODE
109 "**********************************
110 "*
111 " ORG $1500
112 "*
113 "INIT                ;SYS 5376 FROM BASIC
114 "* SET UP IRQVEC
115 " LDA IRQVEC       ;BEEN HERE BEFORE?
116 " BNE INIT1        ;YES, SKIP IT
117 " LDA IRQVEC+1
118 " BNE INIT1        ;ALREADY BEEN HERE
119 " LDA IRQ            ;SAVE IRQ VALUE
120 " STA IRQVEC
121 " LDA IRQ+1
122 " STA IRQVEC+1
123 "* SAVE OLD SYSTEM VECTOR
124 " LDA SYS            ;GET CURRENT VALUE
125 " STA SYSVEC         ;SAVE IT
126 " LDA SYS+1
127 " STA SYSVEC+1
128 "* MAKE SYSTEM VECTOR POINT TO US
129 " LDA #<RESTART      ;GET LSB
130 " STA SYS
131 " LDA #>RESTART      ;GET MSB
132 " STA SYS+1
133 "INIT1
134 " JSR CONFIG         ;GET MODE/SET IRQ
135 " RTS                ;BACK TO BASIC
136 "*
137 "* SYS POINTS HERE >>>>>>>>>>>>
138 "*
139 "RESTART
140 " JSR CONFIG       ;GET MODE/SET IRQ
141 " JMP (SYSVEC)     ;CONTINUE WARM BOOT
142 "CONFIG
143 " LDA MODEFLG              ;READ MODE
144 " AND #$80                 ;BIT 7 ONLY
145 " STA PREVMD               ;SAVE IT
146 " JSR RELAY          ;SET RELAY ON/OFF
147 "* POINT IRQ TO OUR ROUTINE
148 " SEI                ;DISABLE INTR
149 " LDA #<MAIN         ;PT IRQ TO MAIN
150 " STA IRQ
151 " LDA #>MAIN
152 " STA IRQ+1
153 " CLI                      ;ENABLE INTR
154 " RTS                      ;RETURN
155 "**********************************
156 "PREVMD DFB $80      ;PREVIOUS MODE
157 "IRQVEC DA $0000     ;ORIG IRQ VECTOR
158 "SYSVEC DA $0000     ;ORIG SYS VECTOR
159 "*
160 "* IRQ VECTOR POINTS HERE >>>>>>>>
161 "*
162 "MAIN
163 "* CHECK MODE
164 " LDA MODEFLG        ;GET CURRENT MODE
165 " AND #$80           ;BIT 7
166 " CMP PREVMD         ;MODE CHANGE?
167 " STA PREVMD       ;(SAVE CURRENT MODE)
168 " BEQ EXIT           ;NO MODE CHANGE
169 " JSR RELAY          ;YES, SET RELAY
170 "EXIT
171 " JMP (IRQVEC)       ;CONTINUE IRQ
172 "**********************************
173 "* USE PREVMD TO TURN RELAY ON/OFF
174 "RELAY
175 " LDA $0             ;SET DDR TO OUTPUT
176 " ORA #$10           ;SET BIT 4
```

```
177 " STA $0
178 " LDA $1            ;READ I/O PORT
179 " AND #$EF          ;CLEAR BIT 4
180 " STA $1
181 " LDA #1            ;NON-0 TO 192
182 " STA $00C0         ;INTERLOCK BYTE
183 " BIT PREVMD        ;BIT 7=1(80 COL.)?
184 " BMI RLYON         ;YES, TURN RLY ON
185 "RLYOFF             ;NO, TURN IT OFF
186 " LDA $1            ;READ I/O PORT
187 " ORA #$20          ;SET BIT 5
188 " STA $1            ;RELAY OFF
189 " LDA $D011         ;SET SLOW MODE
190 " ORA #$10          ;SET BIT 4
191 " STA $D011         ;UNBLANK VIC
192 " LDX #0
193 " STX $D030         ;SET SLOW SPEED
194 " RTS
195 "RLYON
196 " LDA $1            ;READ I/O PORT
197 " AND #$DF          ;CLEAR BIT 5
198 " STA $1            ;RELAY ON
199 " RTS
```

SCRNSWAP.LOADER

```
·100 REM================================== DE
·110 REM        SCRNSWAP LOADER             CA
·120 REM       RUPERT REPORT #44            DP
·130 REM         FOR C128 ONLY              BL
·140 REM================================== JM
·150 PRINT"LOADING [3"."]"                  AJ
·160 M=5376 :REM $1500 INITIAL ADDRESS << OD
·170 READ B$                                JE
·180 IF B$="XX" THEN 240                    KP
·190 B=DEC(B$)                              EH
·200 CK=CK+B  :REM CHECKSUM                 JG
·210 POKE M,B                               CL
·220 M=M+1                                  PL
·230 GOTO 170                               AA
·240 IF CK<>12298 THEN PRINT"DATA ERROR"
 : STOP                     :REM <<<<<<< NF
·250 PRINT"PROGRAM IS LOADED"               JF
·260 SYS 5376 :REM $1500 START ADDRESS << CD
·270 END                                    IP
·5376 DATA AD, 4E, 15, D0, 27, AD, 4F, 15 MP
·5384 DATA D0, 22, AD, 14, 03, 8D, 4E, 15 AH
·5392 DATA AD, 15, 03, 8D, 4F, 15, AD, 00 GC
·5400 DATA 0A, 8D, 50, 15, AD, 01, 0A, 8D KA
·5408 DATA 51, 15, A9, 30, 8D, 00, 0A, A9 DH
·5416 DATA 15, 8D, 01, 0A, 20, 36, 15, 60 NK
·5424 DATA 20, 36, 15, 6C, 50, 15, A5, D7 AL
·5432 DATA 29, 80, 8D, 4D, 15, 20, 64, 15 LI
·5440 DATA 78, A9, 52, 8D, 14, 03, A9, 15 NB
·5448 DATA 8D, 15, 03, 58, 60, 80, 65, FA IH
·5456 DATA 03, 40, A5, D7, 29, 80, CD, 4D OB
·5464 DATA 15, 8D, 4D, 15, F0, 03, 20, 64 BA
·5472 DATA 15, 6C, 4E, 15, A5, 00, 09, 10 GN
·5480 DATA 85, 00, A5, 01, 29, EF, 85, 01 MH
·5488 DATA A9, 01, 85, C0, 2C, 4D, 15, 30 KE
·5496 DATA 14, A5, 01, 09, 20, 85, 01, AD PD
·5504 DATA 11, D0, 09, 10, 8D, 11, D0, A2 MM
·5512 DATA 00, 8E, 30, D0, 60, A5, 01, 29 AB
·5520 DATA DF, 85, 01, 60                 IL
·5524 DATA XX                             NL
```

BANNER PRINT
FROM PAGE 22

```
·10 GOTO1130                                  FA
·20 IFER=30THENTRAP20:GOTO910                 FL
·30 SLOW:COLOR4,3:COLOR0,3:COLOR5,2:GRAPH
IC0,1:PRINT"ERROR: ";:IFDS>0THENPRINTDS$
:ELSEPRINTERR$(ER):PRINTTAB(7)"IN LINE"E
L                                             LK
·40 PRINT"[DOWN][DOWN]PRESS ANY KEY TO CO
NTINUE":GETKEYA$:COLOR4,1:COLOR0,1:GRAPH
IC1:TRAP20:GOTO940                            KO
·50 COLOR1,2:IFDR=0THEN80                     OG
·60 LOCATECX-24,CY-50:IFRDOT(2)=1THENDRAW
 0,CX-24,CY-50:ELSEDRAW1,CX-24,CY-50          JB
·70 GOTO150                                   GC
·80 COLOR1,2:X=INT((CX-24)/8):Y=INT((CY-5
0)/8):SPRITE1,0:SPRITE2,1,7                   JF
·90 MOVSPR2,X*8+24,Y*8+50:GETKEYA$:IFA$=C
HR$(13)THENSPRITE1,1:SPRITE2,0:GOTO130:E
LSEIFA$=CHR$(20)ORA$="[LEFT]"THENX=X-(1A
NDX>0):GOTO90                                 IN
·100 IFA$="[RIGHT]"THENX=X+(1ANDX<39):GOT
090:ELSEIFA$<" "THEN90                        KP
·110 CHAR1,X,Y,P$+A$,1:X=X+1:IFX=40THENX=
0                                             NP
·120 GOTO90                                   GL
·130 J=JOY(2):GETA$:IFA$<>""THEN220           LH
·140 IFJ>127THENJ=J-128:GOTO310               GL
·150 IFCY<180ANDCY>81THENCX=CX+XX(J):CY=C
Y+YY(J):ELSECX=CX+XX(J)*5:CY=CY+YY(J)*4       CF
·160 IFCX<24THENCX=24:ELSEIFCX>332THENCX=
332                                           EE
·170 IFCY<50THENCY=50:ELSEIFCY>248THENCY=
248                                           EI
·180 IFDR=3THENBEGIN:IFCY<196THENCY=196:E
LSEIFCX<26THENCX=26:ELSEIFCX>262THENCX=2
62                                            BA
·190 BEND                                     NP
·200 MOVSPR1,CX,CY:IFDR>1THENMOVSPR3,CX,C
Y                                             DI
·210 GOTO130                                  OI
·220 IFA$="[UP]"THENCY=CY-8:ELSEIFA$="[DO
WN]"THENCY=CY+8:ELSEIFA$="[LEFT]"THENCX=
CX-8:ELSEIFA$="[RIGHT]"THENCX=CX+8            PP
·230 IFA$=CHR$(13)THENJ=128:GOTO140           KB
·240 IFA$=" "ANDDR=3THEN520                   IB
```

```
·250 IF(A$<"C"ORA$>"W")ORDR>1THEN160            CE
·260 IFA$="D"THEN360:ELSEIFA$="T"THEN370:
 ELSEIFA$="E"THEN380                            LD
·270 IFA$="L"THEN390:ELSEIFA$="H"THEN410:
 ELSEIFA$="M"THEN430                            PL
·280 IFA$="P"THEN530:ELSEIFA$="O"THEN720:
 ELSEIFA$="C"THEN850                            OO
·290 IFA$="G"THEN870:ELSEIFA$="S"THEN890:
 ELSEIFA$="Q"THEN910                            DL
·300 GOTO160                                    FG
·310 IFDR=3THEN490:ELSEIFDR=2THEN460            HJ
·320 IFCY<180ANDCY>81THEN50                     EA
·330 FORI=0TO11:IFCX>TX(I)ANDCX<BX(I)ANDC
 Y>TY(I)ANDCY<BY(I)THEN350                      PF
·340 NEXT:GOTO 160                              CF
·350 ONI+1GOTO360,370,380,390,410,430,530
 ,720,850,870,890,910                           KM
·360 GOSUB970:GOTO130                           OP
·370 GOSUB980:GOTO130                           OC
·380 A$="ERASE BANNER:":GOSUB920:IFRTHENG
 OSUB1060:GOTO130:ELSE130                       GC
·390 COLOR1,16:CHAR1,0,17,"ENTER BANNER L
 ENGTH (1-120) :":N=1:MX=3:X=31:Y=17:GOSU
 B1000:IFZ$=""THEN940                           JM
·400 IFZ<1ORZ>120THEN390:ELSEL=Z:GOSUB106
 0:GOTO940                                      GI
·410 COLOR1,16:CHAR1,0,17,"ENTER BANNER H
 EIGHT (8-24) :":N=1:MX=2:X=28:Y=17:GOSUB
 1000:IFZ$=""THEN940                            DD
·420 IFZ<8ORZ>24THEN410:ELSEH=Z:GOSUB1060
 :GOTO940                                       CL
·430 DR=2:COLOR1,2:CHAR1,2,17,"SELECT ARE
 A TO MAGNIFY & PRESS FIRE"                     JD
·440 FORI=18TO24:CHAR1,0,I,BL$:NEXT             HC
·450 SPRITE1,0:CX=24:CY=81:MOVSPR3,24,81:
 SPRITE3,1,6,,1:GOTO130                         HF
·460 FAST:CHAR1,0,17,BL$:CHAR1,3,17,"PRES
 S SPACE BAR TO RETURN":DR=3:X=CX-24:Y=CY
 -50:OX=X:OY=Y:FORJ=YTOY+13:FORI=XTOX+47        DC
·470 LOCATEI,J:A=(I-X)*5:B=(J-Y)*4+144:IF
 RDOT(2)THENBOX1,A,B,A+4,B+3,,1                 MF
·480 NEXT:NEXT:SPRITE3,0:SLOW:SPRITE1,1:C
 Y=196:MOVSPR1,CX,CY:GOTO130                    EI
·490 X=INT(CX-24):Y=INT(CY-50)                  PK
·500 LOCATEX,Y:A=INT(X/5):B=INT((Y-144)/4
 ):IFRDOT(2)=1THENBOX0,A*5,B*4+144,A*5+4,
 B*4+3+144,,1:C=0:ELSEBOX1,A*5,B*4+144,A*
 5+4,B*4+3+144,,1:C=1                           MJ
·510 DRAWC,OX+A,OY+B:GOTO150                    LM
·520 GOSUB990:GOSUB970:SPRITE3,0:GOTO130        OK
·530 COLOR1,8:CHAR1,0,17,"PREPARE PRINTER
  AND PRESS SPACE BAR"                          MF
·540 GETKEYA$:CHAR1,0,17,BL$:CHAR1,0,17,"
 PRINTING[3"."] SPACE-ABORTS  F-FAST ON/O
 FF"                                            KE
·550 OPEN3,4                                    GB
·560 FP$=CHR$(13):IFFPTHENFP$=CHR$(27)+CH
 R$(10)+CHR$(12)                                CO
·570 IFCT>0THENFORI=1TOCT:PRINT#3,CHR$(CD
 (I));:NEXT                                     LO
·580 IFGR=1THENPRINT#3,CHR$(8);                 FE
·590 FS=0:FORI=40TO120STEP40:Y=(I-40)/40*
 32:IFL>=ITHENX=40:ELSEX=40-(I-L)               DP
·600 IFI>LANDL>40THEN670                        HH
·610 FORC=1TOX*8:FORJ=1TOPW:FORR=HTO1STEP
 -1                                             HJ
·620 GETA$:IFA$<>""THEN690                      GF
·630 LOCATEC-1,R+31+Y:A$=" ":IFGRTHENA$=G
 0$                                             DM
·640 IFRDOT(2)=0THENA$=CH$:IFGRTHENA$=G1$       KD
·650 FORK=1TOPH:PRINT#3,A$;:NEXTK,R:IFGRT
 HENPRINT#3:ELSEPRINT#3,FP$;                    IH
·660 NEXTJ,C,I                                  FI
·670 SLOW:IFGRTHENPRINT#3,CHR$(15);             IK
·680 PRINT#3:CLOSE3:GOTO940                     OC
·690 IFA$=" "THEN670:ELSEIFA$<>"F"THEN630       BP
·700 IFFSTHENSLOW:FS=0:ELSEFAST:FS=1            DB
·710 GOTO630                                    OP
·720 COLOR1,16:CHAR1,0,17,"PRINT USING SO
 LID BLOCKS (Y/N):":N=0:MX=1:X=32:Y=17:GO
 SUB1000                                        KC
·730 CHAR1,0,17,BL$:GR=0:MW=80:IFZ$="Y"TH
 ENGR=1:MW=70:GOTO760                           OK
·740 CHAR1,0,17,"ENTER CHARACTER TO DRAW
 TEXT:":MX=1:X=29:GOSUB 1000:CH$=Z$             NK
·750 CHAR1,0,17,BL$:CHAR 1,0,17,"USE FINE
  LINE SPACING (Y/N):":N=0:MX=1:X=29:Y=17
 :GOSUB1000:FP=0:IFZ$="Y"THENFP=1               OL
·760 CHAR1,0,17,BL$:CHAR1,0,17,"EXPAND TO
  FIT FULL WIDTH OF PAGE(Y/N):":N=0:MX=1:
 X=38:GOSUB1000                                 JK
·770 CHAR1,0,17,BL$:IFZ$="Y"THENPH=INT(MW
 /H):GOTO800                                    EE
·780 CHAR1,0,17,"PIXEL HEIGHT(1-"+STR$(IN
 T(MW/H))+"):":N=1:MX=2:X=21:GOSUB1000:IF
 Z<1ORZ>INT(MW/H)THEN780                        EP
·790 PH=Z:CHAR1,0,17,BL$                        MJ
·800 CHAR1,0,17,"PIXEL WIDTH (1-10):":N=1
 :MX=2:X=20:GOSUB 1000:IFZ<1ORZ>10THEN800       KL
·810 PW=Z:CHAR1,0,17,BL$                        OE
·820 CT=1                                       KI
·830 CHAR1,0,17,BL$:CHAR1,0,17,"SPECIAL P
 RINTER CODE (0=DONE):":N=1:MX=3:X=31:GOS
 UB1000                                         EF
·840 IFZ=0THEN940:ELSECD(CT)=Z:CT=CT+1:IF
 CT<21THEN830:ELSE940                           ED
·850 CHAR1,0,17,"UPPER OR LOWER SET (U/L)
  :":MX=1:N=0:X=27:Y=17:GOSUB 1000:IFZ$="
 U"THENP$=CHR$(142):ELSEIFZ$="L"THENP$=CH
 R$(14)                                         CN
·860 GOTO940                                    IG
·870 CHAR1,0,17,"FILENAME TO LOAD:":MX=12
 :N=0:X=17:Y=17:GOSUB1000:IFZ$=""THEN940        EP
·880 BLOAD(Z$+".BNR"):L=PEEK(9463):H=PEEK
 (9471):POKE9463,0:POKE9471,0:GOTO940           OF
·890 CHAR 1,0,17,"FILENAME TO SAVE AS:":M
```

IMPORTANT! Letters on white background are **Bug Repellent** line codes. **Do not enter them!** Pages 77 and 78 explain these codes and provide other essential information on entering ***Ahoy!*** programs. Refer to these pages **before** entering any programs!

```
X=12:N=0:X=20:Y=17:GOSUB1000:IFZ$=""THEN
940                                          BO
·900 POKE9463,L:POKE9471,H:SCRATCH(Z$+".B
NR"):BSAVE(Z$+".BNR"),B0,P9463TOP12992:P
OKE9463,0:POKE9471,0:GOSUB940:GOTO940        ID
·910 A$="EXIT TO BASIC:":GOSUB920:IFR=0TH
EN130:ELSEGRAPHIC0,1:COLOR0,C0:COLOR4,C4
:COLOR5,C5:SPRITE1,0:SPRITE2,0:SPRITE3,0
:END                                         EM
·920 R=0:CHAR1,0,17,A$+"  ARE YOU SURE? (
Y/N) :":MX=1:N=0:X=24+LEN(A$):Y=17:GOSUB
1000:CHAR1,0,17,BL$:IFZ$="Y"THENR=1          JB
·930 RETURN                                  DD
·940 CHAR1,0,17,BL$:GOTO130                  OK
·950 COLOR1,13:BOX1,X*8-1,Y*8-2,X*8+LEN(A
$)*8+2,Y*8+8                                 EE
·960 COLOR1,C:CHAR1,X,Y,A$,1:COLOR1,2:RET
URN                                          NJ
·970 A$=" DRAW [BACKARROW]":X=X(0):Y=Y(0)
:C=C(0):GOSUB950:DR=1:A$=" TEXT  ":X=X(1
):Y=Y(1):C=C(1):GOSUB950:RETURN              DE
·980 A$=" TEXT [BACKARROW]":X=X(1):Y=Y(1)
:C=C(1):GOSUB950:DR=0:A$=" DRAW  ":X=X(0
):Y=Y(0):C=C(0):GOSUB950:RETURN              BD
·990 FORI=17TO24:CHAR1,0,I,BL$:NEXT:COLOR
1,13:BOX1,1,144,319,198:FORI=0TO11:A$=B$
(I):X=X(I):Y=Y(I):C=C(I):GOSUB950:NEXT:R
ETURN                                        OH
·1000 Z$="":NC=0:CHAR1,X,Y,LEFT$(BL$,40-X
)                                            FM
·1010 GETA$:IFA$=""THENCHAR1,X+NC,Y,"[BAC
KARROW]"                                     PB
·1020 IFA$=CHR$(13)THENCHAR1,X+NC,Y," ":Z
=VAL(Z$):RETURN                              EL
·1030 IFA$=CHR$(20)ANDNC>0THENZ$=LEFT$(Z$
,LEN(Z$)-1):CHAR1,X+NC,Y," ":NC=NC-1:GOT
O1010                                        JI
·1040 IF(A$<" "OR(NAND(A$<"0"ORA$>"9"))OR
LEN(Z$)=MX)THEN 1010                         AP
·1050 Z$=Z$+A$:CHAR1,X+NC,Y,A$:NC=NC+1:GO
TO1010                                       EO
·1060 COLOR1,2:FORI=4TO17:CHAR1,0,I,BL$:N
EXT                                          HI
·1070 IFL<41THENZ=L:ELSEZ=40                 HA
·1080 FORI=1TOH:DRAW1,0,I+31TOZ*8,I+31:NE
XT                                           OA
·1090 IFL<41THENRETURN:ELSEIFL<81THENZ=L:
ELSEZ=80                                     PB
·1100 Z=L-40:FORI=1TOH:DRAW1,0,I+63TOZ*8,
I+63:NEXT                                    OE
·1110 IFL<81THENRETURN                       IJ
·1120 Z=L-80:FORI=1TOH:DRAW1,0,I+95TOZ*8,
I+95:NEXT:RETURN                             JH
·1130 FAST:C0=RCLR(0):C4=RCLR(4):C5=RCLR(
5):TRAP20:DIM S$(23),B$(11),C(11),X(11),
Y(11),TX(11),TY(11),BX(11),BY(11),CD(20)     PI
·1140 COLOR4,1:COLOR0,1:COLOR1,13:GRAPHIC
1,1:FORI=3584TO3775:READJ:POKEI,J:NEXT       OC
·1150 SPRITE1,1,16:MOVSPR1,180,200:CX=180
:CY=200:SPRITE2,0                            AC
·1160 FORI=0TO11:READB$(I),C(I),X(I),Y(I)
:TX(I)=X(I)*8+20:TY(I)=Y(I)*8+47:BX(I)=T
X(I)+LEN(B$(I))*8+9:BY(I)=TY(I)+14:NEXT      HJ
·1170 FORI=1TO7:G0$=G0$+CHR$(128):G1$=G1$
+CHR$(255):READXX(I),YY(I):NEXT:READXX(I
),YY(I)                                      GM
·1180 MW=80:PW=1:PH=1:FP=0:EX=1:GR=0:CH$=
"0":CT=0:L=40:H=8                            IO
·1190 BL$="[40" "]":Q$=CHR$(34)              KN
·1200 FORI=0TO2:CHAR1,0,I,BL$,1:NEXT:BOX0
,5,2,314,21:CHAR1,14,1,"BANNER PRINT",1      GM
·1210 GOSUB990:GOSUB970:GOSUB1060:P$=CHR$
(142):SLOW:GOTO130                           JO
·1220 DATA 255,0,0,254,0,0,252,0             NN
·1230 DATA 0,254,0,0,223,0,0,135             OO
·1240 DATA 128,0,3,192,0,1,224,0             JI
·1250 DATA 0,192,0,0,0,0,0,0                 FF
·1260 DATA 0,0,0,0,0,0,0,0                   BK
·1270 DATA 0,0,0,0,0,0,0,0                   EA
·1280 DATA 0,0,0,0,0,0,0,0                   EJ
·1290 DATA 0,0,0,0,0,0,0,0                   DH
·1300 DATA 255,0,0,129,0,0,129,0             EK
·1310 DATA 0,129,0,0,129,0,0,129             JO
·1320 DATA 0,0,129,0,0,255,0,0               ID
·1330 DATA 0,0,0,0,0,0,0,0                   FP
·1340 DATA 0,0,0,0,0,0,0,0                   IF
·1350 DATA 0,0,0,0,0,0,0,0                   HD
·1360 DATA 0,0,0,0,0,0,0,0                   HJ
·1370 DATA 0,0,0,0,0,0,0,0                   KH
·1380 DATA 255,255,255,128,0,1,128,0         JN
·1390 DATA 1,128,0,1,128,0,1,128             HO
·1400 DATA 0,1,128,0,1,128,0,1               MA
·1410 DATA 128,0,1,128,0,1,128,0             CB
·1420 DATA 1,128,0,1,128,0,1,255             NB
·1430 DATA 255,255,0,0,0,0,0,0               JP
·1440 DATA 0,0,0,0,0,0,0,0                   OJ
·1450 DATA 0,0,0,0,0,0,0,0                   NH
·1460 DATA " DRAW  ",3,1,19," TEXT  ",3,1
,21, " ERASE ",3,1,23                        EC
·1470 DATA " LENGTH  ",6,10,19," HEIGHT
",6,10,21," MAGNIFY ",6,10,23                NO
·1480 DATA " PRINT[3" "]",7,21,19," OPTIO
NS ",7,21,21," CHR SET ",7,21,23             JJ
·1490 DATA " LOAD ",8,33,19," SAVE ",8,33
,21," QUIT ",8,33,23                         FD
·1500 DATA 0,-1,1,-1,1,0,1,1,0,1,-1,1,-1,
0,-1,-1                                      EG
```

STOP TYPING!

All the programs in this issue are available on disk! See page 30.

ROUTE 64
FROM PAGE 13

```
•10 REM *** ROUTE 64 *** BUCK CHILDRESS *
**                                          JG
•20 REM *** P.O. BOX 13575 SALEM, OR 9730
9 ***                                       OA
•30 PRINTCHR$(147)"LOADING";:J=52000        AP
•40 PRINT".";:FORB=0TO11:READA              EA
•50 IFA<0ORA>255THEN70                      EP
•60 POKEJ+B,A:X=X+A:D=D+1:NEXTB:READA:IFA
=XTHEN80                                    GO
•70 PRINT:PRINT:PRINT"ERROR IN DATA LINE:
"PEEK(64)*256+PEEK(63):END                  HB
•80 X=0:J=J+12:IFD<840THEN40                GL
•90 PRINT:PRINT:PRINT"THE DATA IS OK[3"."
]":PRINT                                    LH
•100 PRINT"SYS 52[3"0"] TO ACTIVATE[3"."]
":END                                       NN
•110 DATA169,0,141,102,206,165,43,166,44,
133,251,133,1553                            GI
•120 DATA253,134,252,134,254,32,225,255,2
40,71,32,208,2090                           ID
•130 DATA205,140,103,206,140,104,206,140,
105,206,140,106,1801                        GO
•140 DATA206,140,108,206,32,163,205,208,4
9,173,102,206,1798                          FA
•150 DATA208,43,238,102,206,169,38,160,20
6,32,30,171,1603                            CP
•160 DATA169,0,133,198,133,204,165,203,20
1,63,240,11,1720                            KM
•170 DATA201,25,240,4,201,39,208,242,32,2
10,255,133,1790                             JD
•180 DATA204,162,0,134,207,201,25,240,168
,96,32,171,1640                             DJ
•190 DATA205,177,251,170,32,171,205,177,2
51,32,205,189,2065                          OP
•200 DATA169,32,32,210,255,173,102,206,24
0,10,32,163,1624                            NP
•210 DATA205,201,39,240,50,76,5,204,32,16
3,205,240,1660                              GC
•220 DATA91,142,109,206,72,32,77,205,104,
166,212,208,1624                            EH
•230 DATA239,201,39,240,18,201,137,240,8,
201,141,240,1905                            CJ
•240 DATA4,201,167,208,223,141,108,206,76
,160,203,174,1871                           FC
•250 DATA108,206,240,212,141,104,206,141,
109,206,141,112,1926                        MO
•260 DATA206,169,0,141,99,206,32,163,205,
208,3,76,1508                               CK
•270 DATA104,204,201,34,240,249,32,77,205
,174,105,206,1831                           AH
•280 DATA208,241,201,39,208,232,142,104,2
06,173,102,206,2062                         LG
•290 DATA208,58,141,107,206,76,136,204,17
3,106,206,208,1829                          DN
•300 DATA23,173,103,206,240,7,165,253,166
,254,76,41,1707                             LJ
•310 DATA203,160,1,177,253,170,136,177,25
3,76,41,203,1850                            IA
•320 DATA165,214,201,2,144,35,201,3,176,4
,165,211,1521                               HI
•330 DATA240,27,169,21,160,206,32,30,171,
76,128,204,1464                             LA
•340 DATA32,178,205,32,163,205,240,6,32,7
1,205,76,1445                               BN
•350 DATA69,204,238,103,206,169,222,160,2
05,32,30,171,1809                           EH
•360 DATA169,204,162,5,32,205,189,169,19,
162,13,160,1489                             BL
•370 DATA4,141,119,2,142,120,2,142,121,2,
142,122,1059                                PP
•380 DATA2,132,198,96,32,49,205,169,227,1
60,205,32,1507                              FC
•390 DATA30,171,169,10,160,206,32,30,171,
169,112,160,1420                            FD
•400 DATA206,32,30,171,169,13,32,210,255,
76,210,255,1659                             CP
•410 DATA165,43,166,44,133,63,134,64,160,
0,177,63,1212                               AD
•420 DATA133,65,200,177,63,133,66,162,255
,160,3,142,1559                             GI
•430 DATA100,206,140,101,206,238,100,206,
238,101,206,172,2014                        PG
•440 DATA101,206,177,63,201,128,144,40,56
,233,127,170,1646                           NB
•450 DATA160,255,202,240,8,200,185,158,16
0,16,250,48,1882                            HD
•460 DATA245,200,185,158,160,48,14,174,10
0,206,221,112,1823                          EE
•470 DATA206,208,17,238,100,206,76,197,20
4,56,233,128,1869                           DO
•480 DATA174,100,206,221,112,206,240,21,1
65,65,166,66,1742                           DJ
•490 DATA208,162,173,107,206,208,51,32,49
,205,169,253,1823                           HH
•500 DATA160,205,76,111,204,236,99,206,14
4,167,238,107,1953                          NE
•510 DATA206,173,107,206,201,2,176,16,160
,2,177,63,1489                              EB
•520 DATA141,110,206,200,177,63,141,111,2
06,76,228,204,1863                          FE
•530 DATA32,49,205,169,239,160,205,76,111
,204,32,178,1660                            GK
•540 DATA205,173,111,206,174,110,206,32,2
05,189,76,160,1847                          ID
•550 DATA203,169,0,174,99,206,141,109,206
,232,157,112,1808                           CF
•560 DATA206,32,166,205,240,96,32,163,205
,240,91,32,1708                             MD
•570 DATA77,205,76,66,205,166,212,208,36,
201,128,144,1724                            IM
•580 DATA32,56,233,127,170,160,255,202,24
0,8,200,185,1868                            FB
```

```
•590 DATA158,160,16,250,48,245,200,185,15
8,160,48,6,1634                          JL
•600 DATA32,117,205,76,102,205,56,233,128
,174,109,206,1643                        DN
•610 DATA240,9,238,99,206,174,99,206,157,
112,206,32,1778                          JN
•620 DATA210,255,174,104,206,240,23,201,3
9,240,19,201,1912                        CD
•630 DATA48,144,12,201,58,144,11,201,65,1
44,4,201,1233                            GJ
•640 DATA91,144,3,141,105,206,96,32,171,2
05,162,0,1356                            KJ
•650 DATA161,251,96,230,251,208,2,230,252
,96,169,20,1966                          ED
•660 DATA141,106,206,174,99,206,32,210,25
5,202,16,250,1897                        IF
•670 DATA173,102,206,240,236,160,1,177,25
1,208,230,169,2153                       CA
•680 DATA58,76,210,255,32,68,229,160,0,13
2,198,132,1550                           AD
•690 DATA199,132,212,132,216,96,13,83,89,
83,0,13,1268                             BJ
•700 DATA13,18,69,82,82,79,82,32,73,78,0,
13,621                                   HB
•710 DATA13,18,68,85,80,76,73,67,65,84,69
,68,766                                  KP
•720 DATA0,13,13,18,85,78,68,69,70,73,78,
69,634                                   CI
•730 DATA68,0,32,76,65,66,69,76,32,42,42,
32,600                                   EP
•740 DATA0,13,13,18,76,73,78,69,32,84,79,
79,614                                   MK
•750 DATA32,76,79,78,71,0,17,76,73,78,69,
32,681                                   HG
•760 DATA82,69,70,69,82,69,78,67,69,83,32
,67,837                                  CA
•770 DATA79,77,80,76,69,84,69,46,46,46,13
,13,698                                  HN
•780 DATA82,69,77,79,86,69,32,69,88,73,83
,84,891                                  GE
•790 DATA73,78,71,32,76,65,66,69,76,83,63
,32,784                                  GF
•800 DATA40,89,47,78,41,32,0,0,0,0,0,0,32
7                                        PJ
```

STEEPLECHASE
FROM PAGE 15

Starting address in hex: C000
Ending address in hex: CA80
SYS to start: 49152

Flankspeed **required for entry! See page 79.**

```
C000: A9 8E 20 D2 FF A9 08 20 FC
C008: D2 FF A9 7E 85 39 A9 C7 33
C010: 85 3A A9 00 85 3B A9 3C 20
C018: 85 3C A2 0C A0 00 B1 39 14
C020: 91 3B C8 C0 40 90 F7 18 57
C028: 98 65 39 85 39 A5 3A 69 67
C030: 00 85 3A 18 98 65 3B 85 C6
C038: 3B A5 3C 69 00 85 3C CA 4B
C040: D0 DA A0 18 B9 A9 C5 99 67
C048: 00 D4 88 10 F7 20 C9 C4 5C
C050: A0 2C A9 00 99 19 C5 88 C7
C058: 10 FA 8D 20 D0 8D 15 D0 55
C060: 8D 17 D0 8D 1B D0 8D 1D F9
C068: D0 A9 FF 8D 1C D0 A9 09 10
C070: 8D 25 D0 A9 0F 8D 26 D0 31
C078: A9 07 8D 27 D0 A9 01 8D E6
C080: 28 D0 20 25 C4 A9 92 A0 60
C088: C6 20 1E AB A2 06 A0 00 82
C090: 18 20 F0 FF A9 C2 A0 C5 8C
C098: 20 1E AB A2 12 A0 00 18 EF
C0A0: 20 F0 FF A9 C2 A0 C5 20 A4
C0A8: 1E AB A0 27 A9 A0 99 A0 BE
C0B0: 04 99 E0 05 99 80 06 99 ED
C0B8: C0 07 A9 00 99 A0 D8 99 D6
C0C0: 80 DA A9 05 99 E0 D9 99 B8
C0C8: C0 DB 88 10 DF A2 02 A0 23
C0D0: 03 18 20 F0 FF A9 6B A0 B2
C0D8: C6 20 1E AB A2 0E A0 03 DD
C0E0: 18 20 F0 FF A9 6B A0 C6 86
C0E8: 20 1E AB 18 AD 19 C5 AA 22
C0F0: 69 31 8D 17 04 8A 0A 0A D2
C0F8: 0A AA A0 00 BD 46 C5 99 B1
C100: 20 04 E8 C8 C0 07 90 F4 23
C108: A0 7F AD 1B D4 99 00 3F 9E
C110: 88 10 F7 A9 7D 8D 2E C5 49
C118: A9 DD 8D 2F C5 A2 02 BD 84
C120: 86 C5 9D 38 C5 9D 3B C5 A6
C128: AD 1B D4 29 FE 9D 40 C5 91
C130: 9D 43 C5 CA 10 E9 20 DE 9A
C138: C2 A9 FF 8D 24 C5 8D 25 CE
C140: C5 8D 15 D0 A2 00 8E 36 E0
C148: C5 8E 37 C5 8A 29 0F 4A A6
C150: 4A A8 09 F0 8D F8 07 8D 58
C158: F9 07 A9 10 8D 04 D4 B9 33
C160: A5 C5 F0 08 8D 01 D4 A9 D1
C168: 11 8D 04 D4 A9 02 20 C1 6D
C170: C4 8A 48 20 DE C2 68 AA DC
C178: E8 E0 64 90 C9 A9 20 8D 58
C180: 08 D4 A2 80 8E 0B D4 E8 D7
C188: 8E 0B D4 20 9A C2 A9 01 1F
C190: 85 04 A6 04 BD 24 C5 C9 36
C198: 08 90 0D BD 00 DC 29 10 12
C1A0: D0 0C 9D 22 C5 9D 24 C5 8A
C1A8: 20 09 C3 4C D1 C1 BD 00 33
C1B0: DC 29 0C 4A 4A A8 18 BD D5
C1B8: 1A C5 79 66 C5 85 02 BD 83
C1C0: 1C C5 79 6A C5 C9 2F B0 F5
C1C8: 08 9D 1C C5 A5 02 9D 1A AF
C1D0: C5 BD 20 C5 F0 06 DE 20 30
C1D8: C5 4C 42 C2 38 A9 32 FD 02
C1E0: 1C C5 9D 20 C5 C9 32 F0 33
C1E8: 2E FE 1E C5 BD 1E C5 29 C4
C1F0: 0F D0 05 20 58 C2 A6 04 BB
```

```
C1F8: BD 1E C5 29 7F D0 13 BD E4
C200: 6E C5 BC 70 C5 20 6E C3 79
C208: CD 17 04 90 03 4C 6D C4 03
C210: A6 04 20 A3 C3 A6 04 BD AA
C218: 24 C5 C9 08 90 24 BD 1E 64
C220: C5 29 1F 4A 4A 4A A8 09 BE
C228: F0 DD F8 07 F0 14 9D F8 92
C230: 07 A2 10 8E 04 D4 B9 A5 B0
C238: C5 F0 07 8D 01 D4 E8 8E D0
C240: 04 D4 C6 04 30 03 4C 92 F5
C248: C1 20 A5 C2 20 DE C2 20 74
C250: 01 C4 20 2F C3 4C 8E C1 C5
C258: 8A 0A A8 38 B9 2B C5 E9 62
C260: 01 29 07 99 2B C5 C9 07 EC
C268: D0 2F BD 72 C5 85 3B BD DC
C270: 74 C5 85 3C A2 06 A0 00 B5
C278: B1 3B 48 C8 B1 3B 88 91 7D
C280: 3B C8 C0 27 D0 F5 68 91 2D
C288: 3B 18 A5 3B 69 28 85 3B 0F
C290: A5 3C 69 00 85 3C CA D0 39
C298: DD 60 A0 02 A9 00 99 08 C4
C2A0: DC 88 10 FA 60 AD 0A DC 06
C2A8: AA 29 F0 4A 4A 4A 4A 09 9F
C2B0: 30 8D 06 04 8A 29 0F 09 44
C2B8: 30 8D 07 04 AD 09 DC AA BF
C2C0: 29 F0 4A 4A 4A 4A 09 30 3D
C2C8: 8D 09 04 8A 29 0F 09 30 5F
C2D0: 8D 0A 04 AD 08 DC 29 0F 37
C2D8: 09 30 8D 0C 04 60 A2 07 B9
C2E0: 8A 0A A8 BD 2E C5 99 01 6A
C2E8: D0 BD 36 C5 99 00 D0 AD 8B
C2F0: 10 D0 3D 7E C5 A8 BD 3E F7
C2F8: C5 29 01 F0 05 98 1D 76 0B
C300: C5 A8 8C 10 D0 CA 10 D8 8F
C308: 60 BC 24 C5 38 BD 91 C5 5C
C310: F9 89 C5 9D 2E C5 98 4A CD
C318: 09 F4 9D F8 07 18 BD 22 AB
C320: C5 69 04 9D 22 C5 BD 24 BA
C328: C5 69 00 9D 24 C5 60 AD EC
C330: 1C C5 A2 69 A0 04 20 40 23
C338: C3 AD 1D C5 A2 49 A0 06 1F
C340: 86 3B 84 3C A8 4A 4A 85 85
C348: 02 98 29 03 AA A0 00 C4 1F
C350: 02 F0 08 A9 A0 91 3B C8 2B
C358: 4C 4F C3 BD 93 C5 91 3B 9B
C360: A9 20 C8 C0 0C B0 06 91 08
C368: 3B C8 4C 63 C3 60 85 3B 01
C370: 84 3C A0 03 B1 3B 85 02 49
C378: 88 B1 3B AA A0 00 B1 3B 26
C380: A4 02 C8 C0 3A 90 0C A0 28
C388: 30 E8 E0 3A 90 05 A2 30 25
C390: 18 69 01 48 98 A0 03 91 29
C398: 3B 88 8A 91 3B 68 A0 00 BC
C3A0: 91 3B 60 8A 0A 18 65 04 E3
C3A8: AA A9 03 85 02 BD 38 C5 43
C3B0: D0 03 DE 40 C5 DE 38 C5 46
C3B8: BD 38 C5 C9 50 D0 3C BD 59
C3C0: 40 C5 A8 29 01 F0 34 98 57
C3C8: 4A A8 B9 00 3F 85 03 29 66
C3D0: 03 A8 09 F8 9D FA 07 B9 D7
C3D8: 97 C5 9D 29 D0 AD 1D D0 69
C3E0: 3D 80 C5 A4 03 30 03 1D 5C
C3E8: 78 C5 8D 1D D0 BC 9B C5 C0
C3F0: A5 03 29 40 D0 01 A8 98 16
C3F8: 9D 30 C5 E8 C6 02 D0 AD BC
C400: 60 AD 1E D0 29 03 F0 1C 36
C408: 85 02 A2 00 46 02 90 0F 1A
C410: A9 0A DD 1C C5 B0 08 9D D9
C418: 1C C5 A9 00 9D 1A C5 E8 0A
C420: E0 02 90 E8 60 A9 93 20 3A
C428: D2 FF E6 02 A5 02 8D 86 9F
C430: 02 A9 C1 A0 C6 20 1E AB EE
C438: AD 00 DC 29 0C 4A 4A A8 35
C440: 18 AD 19 C5 79 A1 C5 C9 8F
C448: 03 B0 03 8D 19 C5 AD 19 32
C450: C5 0A A8 A9 01 99 0E D9 F4
C458: A9 03 20 C1 C4 AD 00 DC 36
C460: 29 10 D0 C6 A9 0E 8D 27 9D
C468: C5 8D 29 C5 60 A0 00 A5 51
C470: 04 0A AA 98 29 03 9D 27 B2
C478: C5 0A 09 19 8D 08 D4 A2 77
C480: 10 8E 0B D4 E8 8E 0B D4 56
C488: A9 09 20 C1 C4 C8 C0 14 7F
C490: 90 DD A0 00 B9 20 04 D9 57
C498: 06 04 F0 05 B0 0B 4C 50 F0
C4A0: C0 C8 C0 07 90 EE 4C 50 0E
C4A8: C0 AD 19 C5 0A 0A 0A AA BE
C4B0: A0 00 B9 06 04 9D 46 C5 BE
C4B8: E8 C8 C0 07 90 F4 4C 50 54
C4C0: C0 18 65 A2 C5 A2 D0 FC D7
C4C8: 60 78 A9 7F 8D 0D DC A9 EB
C4D0: 01 8D 1A D0 8D 12 D0 A9 64
C4D8: 1B 8D 11 D0 A9 E8 8D 14 97
C4E0: 03 A9 C4 8D 15 03 58 60 B0
C4E8: A9 01 8D 19 D0 A0 03 AD 5C
C4F0: 12 D0 D9 5E C5 B0 03 88 0E
C4F8: D0 F8 B9 26 C5 8D 21 D0 E7
C500: B9 2A C5 8D 16 D0 B9 62 3A
C508: C5 8D 12 D0 AD 0D DC 29 FE
C510: 01 F0 03 4C 31 EA 4C BC 76
C518: FE 00 00 00 00 00 00 00 17
C520: 00 00 00 00 00 00 00 00 20
C528: 00 00 00 00 00 00 00 00 28
C530: 00 00 00 00 00 00 00 00 30
C538: 00 00 00 00 00 00 00 00 38
C540: 00 00 00 00 00 00 64 64 09
C548: 3A 64 64 3A 64 00 64 64 B2
C550: 3A 64 64 3A 64 00 64 64 BA
C558: 3A 64 64 3A 64 00 00 53 4D
C560: 92 B5 53 92 B5 00 FD 0A 4C
C568: F0 FD FF 00 FF FF 5C 3C EF
C570: 04 06 F0 D0 04 06 01 02 49
C578: 04 08 10 20 40 80 FE FD 72
C580: FB F7 EF DF BF 7F 00 90 14
C588: FF 03 07 0A 0C 0A 07 03 BC
C590: 00 7D DD 20 65 61 E7 05 BF
```

```
C598: 0C 00 05 84 84 84 E4 E4 01
C5A0: E4 00 01 FF 00 04 08 00 92
C5A8: 04 00 00 00 00 00 01 00 AD
C5B0: 00 00 00 00 00 19 09 FF D2
C5B8: FF 00 00 80 00 00 00 00 39
C5C0: 00 8F 05 20 20 20 A4 AF 0A
C5C8: B9 A2 B9 AF A4 20 20 20 93
C5D0: 20 20 20 20 20 20 20 20 D1
C5D8: 20 A4 AF B9 A2 AF A4 20 1E
C5E0: 20 20 20 20 20 20 20 20 E1
C5E8: 20 20 0D 20 20 A3 B7 B8 8A
C5F0: B7 B8 B7 A3 20 20 A4 AF 51
C5F8: B9 A2 B9 AF A4 20 20 20 C3
C600: A3 B7 B8 B8 B7 A3 20 20 68
C608: 20 20 A4 AF B9 A2 B9 AF 62
C610: A4 20 0D 20 20 20 20 20 82
C618: 20 20 20 20 20 20 A3 B7 34
C620: B8 B7 A3 B7 A3 20 20 20 EF
C628: 20 20 20 20 20 20 20 20 29
C630: 20 20 20 20 A3 B7 B8 B8 7D
C638: B7 A3 0D 0D 0D 9C A4 AF AB
C640: B9 A2 12 B8 B7 A3 20 B7 9A
C648: B8 92 B9 AF A4 AF B9 A2 AD
C650: B9 AF A4 20 20 A4 AF B9 AC
C658: A2 12 B8 B7 A3 B7 B7 B8 49
C660: 92 B9 AF A4 20 A4 AF B9 2F
C668: AF A4 00 96 44 49 53 54 88
C670: 41 4E 43 45 20 05 30 2E 0C
C678: 30 30 20 20 20 9C 53 50 79
C680: 45 45 44 20 99 20 20 20 69
C688: 20 20 20 20 20 20 20 20 89
C690: 20 00 93 20 9A 54 49 4D E9
C698: 45 20 05 30 30 3A 30 30 FD
C6A0: 3A 30 20 20 20 9F 43 4F 9D
C6A8: 55 52 53 45 20 05 20 20 4E
C6B0: 20 20 9E 42 45 53 54 20 DE
C6B8: 05 20 20 20 20 20 20 20 9E
C6C0: 00 13 11 20 20 D5 C3 C9 88
C6C8: B0 B2 AE B2 C3 AE B2 C3 76
C6D0: AE B2 C3 C9 B2 20 20 B2 65
C6D8: C3 AE D5 C3 C9 B2 20 B2 34
C6E0: D5 C3 C9 D5 C3 C9 B2 C3 1E
C6E8: AE 0D 20 20 CA C3 C9 20 5D
C6F0: C2 20 AB B3 20 AB B3 20 D2
C6F8: AB C3 CB C2 20 20 AB B3 96
C700: 20 C2 20 20 AB C3 B3 AB F1
C708: C3 B3 CA C3 C9 AB B3 0D 44
C710: 20 20 CA C3 CB 20 B1 20 9C
C718: B1 C3 BD B1 C3 BD B1 20 50
C720: 20 B1 C3 BD B1 C3 BD CA 71
C728: C3 CB B1 20 B1 B1 20 B1 BE
C730: CA C3 CB B1 C3 BD 0D 0D D7
C738: 0D 1F 20 20 20 20 53 45 7D
C740: 4C 45 43 54 20 4C 45 4E 69
C748: 47 54 48 20 4F 46 20 43 45
C750: 4F 55 52 53 45 20 20 28 48
C758: 31 20 32 20 33 29 0D 0D 72
C760: 20 20 20 20 20 20 20 20 61
C768: 20 20 20 1C 50 52 45 53 20
C770: 53 20 46 49 52 45 20 42 6D
C778: 55 54 54 4F 4E 00 00 00 14
C780: 00 00 00 00 00 00 00 00 80
C788: 00 00 00 00 00 00 04 00 8C
C790: 00 05 00 00 0C 00 00 14 B5
C798: 00 00 50 10 00 D4 68 01 37
C7A0: F1 A9 09 7E 66 6A 5D 92 84
C7A8: 6A 9E 40 6A 65 80 29 56 C1
C7B0: 90 29 00 90 24 00 24 84 C7
C7B8: 00 20 80 00 08 26 00 00 87
C7C0: 00 00 00 00 00 00 00 00 C0
C7C8: 00 00 00 04 00 00 05 00 D1
C7D0: 00 0C 00 00 14 10 00 50 51
C7D8: 68 00 D5 A9 01 FD A6 09 6F
C7E0: 5E 52 6A 5D 80 6A 95 40 1A
C7E8: 6A 66 80 29 56 90 09 00 53
C7F0: 90 08 00 90 09 01 80 02 A6
C7F8: 02 00 00 00 00 26 00 00 21
C800: 00 00 00 00 00 00 00 00 00
C808: 04 00 00 05 00 00 0C 00 1D
C810: 00 14 10 00 50 68 00 D5 C2
C818: A9 00 F1 A6 01 3E 52 09 F4
C820: 5D 80 6A 5E 40 6A 95 80 87
C828: 6A 66 80 69 56 40 09 02 84
C830: 40 0A 02 00 02 48 00 00 C6
C838: 80 00 00 00 00 26 00 00 DE
C840: 00 00 00 00 00 00 00 00 40
C848: 00 00 00 04 00 00 05 00 51
C850: 00 0C 00 00 14 10 00 50 D0
C858: 68 00 D5 A9 01 F1 A6 09 E2
C860: 7E 52 6A 5D 80 6A 9E 40 C2
C868: 6A 65 80 29 56 90 09 00 D1
C870: 90 08 00 90 09 01 80 02 26
C878: 02 00 00 00 00 26 00 00 A0
C880: 00 00 10 00 00 14 40 00 E4
C888: 31 A0 00 51 A4 00 46 98 2F
C890: 01 56 98 01 C5 10 04 F6 52
C898: 40 05 79 00 09 7A 00 2A 05
C8A0: 56 40 6A 98 90 6A 50 90 16
C8A8: 6A 40 80 09 02 00 09 00 E7
C8B0: 00 09 00 00 08 40 00 02 04
C8B8: 00 00 00 00 00 00 00 04 BC
C8C0: 00 00 05 00 00 0C 00 00 D1
C8C8: 14 10 00 50 68 00 55 A9 A4
C8D0: 00 F1 A6 01 3D 86 09 5E 95
C8D8: 50 2A 5E 80 6A 95 80 6A 1D
C8E0: 66 90 6A 52 10 09 02 10 BF
C8E8: 0A 48 40 02 50 00 00 80 4E
C8F0: 00 00 00 00 00 00 00 00 F0
C8F8: 00 00 00 00 00 26 00 10 2F
C900: 00 00 14 00 00 30 00 00 44
C908: 50 00 00 40 00 00 54 10 FC
C910: 00 F4 68 01 7D A9 0A 5D FC
C918: A6 6A 9E 66 6A A5 90 6A 39
C920: 66 80 0A 56 40 02 40 90 7A
C928: 02 10 90 00 84 84 00 22 F5
C930: 00 00 00 00 00 00 00 00 30
```

```
C938: 00 00 00 00 00 26 00 00 5E
C940: 00 00 00 00 00 40 00 00 80
C948: 50 00 00 C0 00 00 40 00 99
C950: 01 40 00 01 50 10 09 F4 F0
C958: 68 2A 7C A9 6A 9D A6 6A 2A
C960: 9E 52 6A 5E 90 09 15 80 49
C968: 09 06 40 02 42 40 02 40 7E
C970: 90 00 00 90 00 00 24 00 B5
C978: 00 00 00 00 00 26 00 00 9E
C980: 00 00 00 00 00 00 00 00 80
C988: CB 00 00 3A 20 03 AB 8C E9
C990: 02 BA 68 0A A6 93 3B 6A 9F
C998: 60 09 9E 6C 26 69 A8 0E 53
C9A0: 99 9B 09 A5 6C 02 55 90 D8
C9A8: 0E 24 B0 00 00 00 00 00 8B
C9B0: 00 00 00 00 00 00 00 00 B0
C9B8: 00 00 00 00 00 26 01 55 35
C9C0: 40 03 FB C0 02 EF 80 03 36
C9C8: BE C0 03 BB 80 02 FB C0 46
C9D0: 03 AF 80 03 EE C0 02 BB 74
C9D8: 80 03 EE C0 03 BE C0 02 90
C9E0: ED 80 03 76 C0 06 D5 D0 36
C9E8: 15 55 54 00 00 00 00 00 A7
C9F0: 00 00 00 00 00 00 00 00 F0
C9F8: 00 00 00 00 00 26 00 00 1F
CA00: 00 00 00 00 00 00 00 10 10
CA08: 00 10 55 55 54 9A AA 98 F4
CA10: 10 00 10 10 00 10 55 55 FA
CA18: 54 9A AA 98 10 00 10 10 7A
CA20: 00 10 55 55 54 9A AA 98 0D
CA28: 10 00 10 00 00 00 00 00 48
CA30: 00 00 00 00 00 00 00 00 30
CA38: 00 00 00 00 00 26 00 00 5E
CA40: 00 00 00 00 00 00 00 00 40
CA48: 00 00 00 00 00 00 00 00 48
CA50: 00 00 00 00 00 00 00 00 50
CA58: 00 00 00 00 00 14 00 00 6C
CA60: 69 40 01 9A 90 16 B7 E4 E8
CA68: 6A FD F9 03 F4 00 00 00 C2
CA70: 00 00 00 00 00 00 00 00 70
CA78: 00 00 00 00 00 26 00 00 9E
CA80: 00 80
```

SCANNER

FROM PAGE 25

Starting address in hex: C000
Ending address in hex: C588
SYS to start: 49152

Flankspeed **required for entry! See page 79.**

```
C000: 20 18 E5 A9 8E 20 D2 FF 49
C008: A9 08 20 D2 FF A9 0C 8D EF
C010: 20 D0 A9 0F 8D 21 D0 A9 E2
C018: 08 A2 D8 85 8B 86 8C A2 62
C020: 05 20 7E C3 20 8D C3 20 19
C028: 8D C3 20 8D C3 20 7E C3 4D
C030: CA D0 EE A9 08 A2 04 85 98
C038: 8B 86 8C A2 05 A0 00 B9 D8
C040: 02 C5 91 8B C8 C0 19 D0 98
C048: F6 20 9D C3 20 AB C3 20 70
C050: AB C3 20 AB C3 A0 00 B9 A9
C058: 34 C5 91 8B C8 C0 19 D0 E2
C060: F6 20 9D C3 CA D0 D6 A9 F4
C068: 00 85 FB 85 FC A9 7A A2 33
C070: 07 85 FD 86 FE A9 00 8D B7
C078: C3 C5 8D C2 C5 8D C3 C5 2F
C080: 8D C4 C5 A9 E1 A2 C0 8D 15
C088: 18 03 8E 19 03 A9 7F 8D 05
C090: 0D DD A9 82 8D 0D DD A9 C9
C098: FF A2 0F 8D 04 DD 8E 05 4D
C0A0: DD A9 11 8D 0E DD A9 10 6C
C0A8: A2 00 8D 06 DD 8E 07 DD 30
C0B0: A9 51 8D 0F DD A5 91 C9 27
C0B8: 7F D0 06 20 39 C1 4C 66 DC
C0C0: FE A5 C5 C9 05 D0 06 20 F0
C0C8: 57 C1 4C B5 C0 C9 04 D0 43
C0D0: 06 20 7F C1 4C B5 C0 C9 C4
C0D8: 03 D0 DA 20 BB C3 4C B5 29
C0E0: C0 48 8A 48 98 48 A9 00 47
C0E8: 8D C0 C5 20 3F C1 20 AA E8
C0F0: C1 20 09 C2 20 34 C2 A5 5B
C0F8: FD 38 E9 C8 85 FD A5 FE 0A
C100: E9 00 85 FE A5 FB 18 69 91
C108: 01 85 FB A5 FC 69 00 85 1C
C110: FC EE C0 C5 AD C0 C5 C9 80
C118: 05 D0 D0 A9 7A A2 07 85 12
C120: FD 86 FE A5 FB 38 E9 05 6C
C128: 85 FB A5 FC E9 00 85 FC B8
C130: 4C 51 FE A9 82 8D 0D DD 71
C138: 60 A9 7F 8D 0D DD 60 A0 3B
C140: 00 A5 FC AA 20 09 C2 88 02
C148: 88 A5 FB AA 20 09 C2 A0 A9
C150: 00 B1 FB AA A0 06 60 20 CF
C158: 39 C1 20 4C C2 20 4C C2 B1
C160: 20 4C C2 20 4C C2 20 4C 2B
C168: C2 20 44 C3 20 3F C1 20 94
C170: AA C1 20 09 C2 20 34 C2 DF
C178: 20 33 C1 20 DF C4 60 20 D2
C180: 39 C1 20 A5 C2 20 A5 C2 8C
C188: 20 A5 C2 20 52 C3 20 3F A6
C190: C1 20 AA C1 20 09 C2 20 EA
C198: 34 C2 20 68 C3 20 A5 C2 64
C1A0: 20 A5 C2 20 33 C1 20 DF 3E
C1A8: C4 60 8A 29 80 4A 4A 4A E0
C1B0: 4A 4A 4A 4A 09 B0 91 FD 23
C1B8: C8 8A 29 40 4A 4A 4A 4A 9E
C1C0: 4A 4A 09 B0 91 FD C8 8A F1
C1C8: 29 20 4A 4A 4A 4A 4A 09 8E
C1D0: B0 91 FD C8 8A 29 10 4A E7
C1D8: 4A 4A 4A 09 B0 91 FD C8 C9
C1E0: 8A 29 08 4A 4A 4A 09 B0 35
C1E8: 91 FD C8 8A 29 04 4A 4A 8D
C1F0: 09 B0 91 FD C8 8A 29 02 B8
C1F8: 4A 09 B0 91 FD C8 8A 29 09
C200: 01 09 B0 91 FD C8 C8 C8 A4
```

```
C208: 60 8A 29 F0 4A 4A 4A 4A 36
C210: 09 30 C9 3A 90 03 38 E9 03
C218: 39 18 69 80 91 FD C8 8A 36
C220: 29 0F 09 30 C9 3A 90 03 29
C228: 38 E9 39 18 69 80 91 FD 15
C230: C8 C8 C8 60 8A C9 20 90 EF
C238: 0E C9 60 B0 0A C9 40 90 C5
C240: 08 38 E9 40 4C 49 C2 A9 AC
C248: 20 91 FD 60 A0 00 B9 08 BA
C250: 04 99 8E C5 B9 08 D8 99 76
C258: A7 C5 C8 C0 19 D0 EF A9 D2
C260: 03 8D 75 C2 8D 78 C2 A9 9B
C268: D7 8D 7B C2 8D 7E C2 A2 7D
C270: 04 A0 40 B9 E8 07 99 C0 59
C278: 07 B9 E8 DB 99 C0 DB C8 FC
C280: D0 F1 EE 75 C2 EE 78 C2 94
C288: EE 7B C2 EE 7E C2 CA D0 81
C290: E2 A0 00 B9 8E C5 99 C8 84
C298: 07 B9 A7 C5 99 C8 DB C8 CD
C2A0: C0 19 D0 EF 60 A0 00 B9 F5
C2A8: C8 07 99 8E C5 B9 C8 DB C4
C2B0: 99 A7 C5 C8 C0 19 D0 EF 1B
C2B8: A9 07 8D CE C2 8D D1 C2 AA
C2C0: A9 DB 8D D4 C2 8D D7 C2 93
C2C8: A2 04 A0 BF B9 00 07 99 2A
C2D0: 28 07 B9 00 DB 99 28 DB 33
C2D8: 88 D0 F1 E0 04 D0 0F AD 96
C2E0: 00 07 8D 28 07 AD 00 DB 2E
C2E8: 8D 28 DB 4C 20 C3 E0 03 8E
C2F0: D0 0F AD 00 06 8D 28 06 40
C2F8: AD 00 DA 8D 28 DA 4C 20 7E
C300: C3 E0 02 D0 0F AD 00 05 39
C308: 8D 28 05 AD 00 D9 8D 28 FF
C310: D9 4C 20 C3 AD 00 04 8D 59
C318: 28 04 AD 00 D8 8D 28 D8 59
C320: 88 CE CE C2 CE D1 C2 CE 3B
C328: D4 C2 CE D7 C2 CA D0 9C 61
C330: A0 00 B9 8E C5 99 08 04 84
C338: B9 A7 C5 99 08 D8 C8 C0 63
C340: 19 D0 EF 60 38 A5 FB E9 3E
C348: 01 85 FB A5 FC E9 00 85 DC
C350: FC 60 18 A5 FB 69 05 85 5B
C358: FB A5 FC 69 00 85 FC A9 8C
C360: 0A A2 04 85 FD 86 FE 60 7A
C368: 38 A5 FB E9 04 85 FB A5 57
C370: FC E9 00 85 FC A9 7A A2 A0
C378: 07 85 FD 86 FE 60 A0 00 89
C380: A9 00 91 8B C8 C0 19 D0 BA
C388: F9 20 9D C3 60 A0 00 B9 BE
C390: 4D C5 91 8B C8 C0 19 D0 34
C398: F6 20 9D C3 60 18 A5 8B BA
C3A0: 69 28 85 8B A5 8C 69 00 DE
C3A8: 85 8C 60 A0 00 B9 1B C5 56
C3B0: 91 8B C8 C0 19 D0 F6 20 58
C3B8: 9D C3 60 A2 00 BD 08 04 E6
C3C0: 9D 8E C5 BD 08 D8 9D A7 96
C3C8: C5 E8 E0 19 D0 EF A2 00 D4
C3D0: A9 0C 9D 00 D8 E8 E0 28 EE
C3D8: D0 F8 A9 00 8D 1D D8 8D 5D
C3E0: 1E D8 8D 1F D8 8D 20 D8 E3
C3E8: 8D 21 D8 A2 00 BD 66 C5 FC
C3F0: 9D 00 04 E8 E0 28 D0 F5 4B
C3F8: A9 12 20 D2 FF A9 90 20 02
C400: D2 FF 20 85 C4 8D C4 C5 55
C408: 18 A2 00 A0 1E 20 F0 FF 92
C410: AD C5 C5 20 D2 FF C8 18 1D
C418: 20 F0 FF A9 A4 20 D2 FF 6A
C420: 20 85 C4 8D C3 C5 18 A2 5C
C428: 00 A0 1F 20 F0 FF AD C5 6C
C430: C5 20 D2 FF C8 18 20 F0 DA
C438: FF A9 A4 20 D2 FF 20 85 1F
C440: C4 8D C2 C5 18 A2 00 A0 76
C448: 20 20 F0 FF AD C5 C5 20 D2
C450: D2 FF C8 18 20 F0 FF A9 BE
C458: A4 20 D2 FF 20 85 C4 8D E7
C460: C1 C5 A2 00 A9 20 9D 00 F1
C468: 04 E8 E0 28 D0 F8 A2 00 CA
C470: BD A7 C5 9D 08 D8 BD 8E 66
C478: C5 9D 08 04 E8 E0 19 D0 9B
C480: EF 20 BC C4 60 A2 00 A9 BE
C488: 00 9D 77 02 E8 E0 0A D0 44
C490: F8 85 C6 20 E4 FF C9 00 A4
C498: F0 F9 C9 30 90 E7 C9 47 07
C4A0: B0 E3 C9 3A 90 06 C9 41 DA
C4A8: B0 0B 90 D9 8D C5 C5 38 20
C4B0: E9 30 4C BB C4 8D C5 C5 B0
C4B8: 38 E9 37 60 AD C2 C5 0A B2
C4C0: 0A 0A 0A 18 6D C1 C5 AA 96
C4C8: AD C4 C5 0A 0A 0A 0A 18 41
C4D0: 6D C3 C5 A8 20 39 C1 86 12
C4D8: FB 84 FC 20 33 C1 60 A2 6E
C4E0: E6 A0 FF 88 D0 FD CA D0 5B
C4E8: F8 60 53 43 41 4E 4E 45 FB
C4F0: 52 20 4A 4F 48 4E 20 4B FE
C4F8: 52 55 54 43 48 20 31 39 0B
C500: 38 37 20 64 64 64 64 64 85
C508: 64 64 64 64 64 64 64 64 2B
C510: 64 64 64 64 64 64 64 64 33
C518: 64 64 20 67 A0 A0 A0 A0 EA
C520: A0 A0 A0 A0 A0 A0 A0 A0 25
C528: A0 A0 A0 A0 A0 A0 A0 A0 2D
C530: A0 A0 A0 61 20 E2 E2 E2 3C
C538: E2 E2 E2 E2 E2 E2 E2 E2 4F
C540: E2 E2 E2 E2 E2 E2 E2 E2 57
C548: E2 E2 E2 E2 7E 00 06 06 5E
C550: 06 06 06 06 0E 0E 0E 0E A0
C558: 0E 0E 0E 0E 0E 0E 0E 0E C8
C560: 0E 0E 0E 0E 0E 00 A0 A0 E7
C568: A0 A0 A0 A0 A0 A0 A0 A0 6D
C570: 85 8E 94 85 92 A0 8E 85 E5
C578: 97 A0 81 84 84 92 85 93 E6
C580: 93 BA A0 A4 E4 A0 A0 A0 DA
C588: A0 29
```

Problems entering a program? Call 212-239-6089 (if busy or no answer after three rings, 212-239-0855)

IMPORTANT! Letters on white background are **Bug Repellent** line codes. **Do not enter them!** Pages 77 and 78 explain these codes and provide other essential information on entering ***Ahoy!*** programs. Refer to these pages **before** entering any programs!

ARCHER
FROM PAGE 60

FROM PAGE 60

```
·10 SCNCLR:TRAP990:COLOR.,1:COLOR4,1:GRAP
HIC1,1:GRAPHIC.:CHAR.,12,10,"[BLUE]PLEAS
E STAND BY!"                                  CM
·20 RESTORE1040:GOSUB1020:CHAR.,10,12,"PA
TIENCE IS A VIRTUE."                          GJ
·30 RESTORE1250:A=12288:B=12288+22*64-1:F
ORC=ATOB:READD:POKEC,D:NEXT:CHAR.,3,14,"
'GOOD THINGS COME TO HE WHO WAITS[3"."]'
"                                             GJ
·40 RESTORE3020:A=3072:B=3775:FORC=ATOB:R
EADD:POKEC,D:NEXT:POKE4590,35:POKE4591,5
0:POKE4592,63:POKE4593,210:POKE4594,1:PO
KE4599,.:POKE4600,5:POKE4601,4:SYS3072        EO
·50 A=(214*64)+3*6:B=214*64+3*6+23:FORC=A
TOB:POKEC,255:POKEC+64,255:NEXT:A=216*64
+1:B=217*64-1:FORC=ATOBSTEP3:POKEC,255:P
OKEC+64,255:NEXT                              GA
·60 POKE53265,11:PRINTCHR$(27)"M"CHR$(11)
CHR$(142)CHR$(147);:FORX=.TO39:A$=A$+CHR
$(160):NEXT                                   HK
·70 COLOR5,7:FORX=.TO24:PRINTA$:NEXT:POKE
4599,.:POKE54290,17:X=FRE(1)                  PI
·80 PRINT"[HOME][4"[DOWN]"]"TAB(9)"[YELLO
W]AHOY! MAGAZINE PRESENTS"                    CF
·90 PRINT"[HOME][8"[DOWN]"]"TAB(8)"[c 6][
s N][s M][3" "][s O][c Y][c Y][s P] [s O
][c Y][c Y] [c H] [c N] [s O][c Y][c Y]
[s O][c Y][c Y][s P]"                         GD
·100 PRINTTAB(7)"[s N]  [s M]  [s L][c P]
[c P][s @] [c H][3" "][s L][c P][s @] [s
 L][c P][c P] [s L][c P][c P][s @]"           CC
·110 PRINTTAB(6)"[s N][4"[c Y]"][s M] [c
H]  [s M] [s L][c P][c P] [c H] [c N] [s
 L][c P][c P] [c H]  [s M][DOWN]"             OE
·120 PRINTTAB(5)"[3"[s M]"]"SPC(6)"[c 4]F
OR THE C128"SPC(8)"[c 6][RVSON][c *]"         CD
·130 PRINTTAB(6)"[RVSON][3"[c *]"][RVSOFF
][25"[c @]"][RVSON] [c *]"                    IC
·140 PRINTTAB(6)"[3"[sEP]"][25"[c Y]"][RV
SON] [RVSOFF][sEP]"                           PM
·150 PRINTTAB(5)"[3"[s N]"]"SPC(3)"[c 4]B
Y CLEVE BLAKEMORE"SPC(5)"[c 6][sEP][DOWN
][DOWN]":PRINTTAB(7)"[3"[DOWN]"]PRESS FI
RE BUTTON TO BEGIN":POKE53265,27              MF
·160 X=1:J=.:DO:POKE8192+8*96+X,96:POKE81
92+8*96+J,.:X=X+1+(X=7)*8:J=J+1+(J=7)*8:
LOOPUNTILJOY(2)>127:SCNCLR                    KK
·170 REM INITIALIZE VARIABLES                 LB
·180 CLR:DIM J,A,B,R,P,S,E,U,W,Z,G,I,V,T,
H,X,P(8),B(7),N,L,K,SC,X(2),Y(2),OX,OY,A
X,AY,DX(8),DY(8),JD,SP,SF,SM,UR,C$(10),B
M(9),RD(9),IX(6),IY(6)                        AH
·190 L=.:SC=.:P=2040:K=3:V=1:T=2:H=3:X=8:
N=53269:B=191:JD=127:SP=32:SM=1024:UR=40
:SF=35                                        MP
·200 RESTORE800:FORJ=.TO7:B(J)=2[UPARROW]
J:NEXT:FORJ=1TO8:READP(J):NEXT:FORJ=1TO1
0:READC$(J):NEXT:FORJ=1TO8:READDX(J),DY(
J):NEXT:FORJ=3TO6:READIX(J),IY(J):NEXT        BP
·210 B$="[DOWN][9"[LEFT]"]":F$="[6"#"]":P
UDEF"0"                                       LK
·220 T$="[RVSON][BLACK][7" "][RVSOFF]":S$
="[c +][s C][s C][c +][DOWN][4"[LEFT]"][
c +][RVSON][][RVSOFF][c +][DOWN][4"[LEFT
]"][c +][RVSON]()[RVSOFF][c +][DOWN][4"[
LEFT]"][4"[c +]"]":TR$="364436[3"4"]9"        MM
·230 DEFFNA(X)=JAND15:DEFFNS(J)=B+A:DEFFN
D(J)=(A-V)*45:DEFFNW(J)=(RND(V)*E)+G          FJ
·240 DEFFNM(J)=INT(RND(V)*U)*W+Z:DEFFNC(J
)=(JANDH):DEFFNX(J)=RSPPOS(V,.):DEFFNY(J
)=RSPPOS(V,V):DEFFNB(J)=(B(A-V)ANDJ):DEF
FNP(J)=PEEK(SM+X(J)+Y(J)*UR)                  KG
·250 DEFFNPD(J)=PEEK(SM+X(.)+DX(A)+(DY(A)
+Y(.))*UR):DEFFNK(J)=(FNP(J)<>SPANDFNP(J
)<>SF):DEFFNCP(J)=(X(.)=X(R)ANDY(.)=Y(R)
):DEFFNSC(J)=36+(PEEK(2047)-210)              OE
·260 REM ENTRY POINT FOR NEW DUNGEON          DG
·270 L=L+V:IFL=11THENGOSUB900:RUN60:ELSEF
ORA=VTO9:BM(A)=253:NEXT:SC$="":IFL=5THEN
K=K+V:CHAR.,.,24,C$(L):GOSUB840:SOUNDV,3
0000,UR,.,.,5000,5000,T:SLEEP3                PD
·280 GOSUB750:FORJ=.TOX:DO:X(.)=INT(RND(V
)*30)+X:Y(.)=INT(RND(V)*21)+T:LOOPWHILEF
NP(.)<>SP:CHAR.,X(.),Y(.),"[YELLOW][RVSO
N]'":NEXT:PRINTCHR$(146)                      FH
·290 FORJ=VTOT:DO:X(J)=INT(RND(V)*30)+X:Y
(J)=INT(RND(V)*21)+T:LOOPWHILEFNP(J)<>SP
:CHAR.,X(J),Y(J),"[BLACK]*":NEXT:X(.)=T:
Y(.)=T                                        LK
·300 CHAR.,X(.),Y(.),"[GREEN]#":CHAR.,X(V
),Y(V),"[RED]*":CHAR.,X(T),Y(T),"[RED]*"
:POKE4599,V:POKE54290,65                      IC
·310 REM DUNGEON OVERVIEW LOOP                GI
·320 R=V:DO:OX=X(.):OY=Y(.):J=JOY(T):A=FN
A(.):IFATHENX(.)=X(.)+DX(A):Y(.)=Y(.)+DY
(A):IFFNP(.)=SPTHENCHAR.,X(.),Y(.),"[GRE
EN]#":CHAR.,OX,OY," ":ELSEX(.)=OX:Y(.)=O
Y                                             LH
·330 AX=X(R):X(R)=X(R)+(X(R)<X(.))*-V+(X(
R)>X(.))*V:IFFNK(R)THENX(R)=AX                LH
·340 AY=Y(R):Y(R)=Y(R)+(Y(R)<Y(.))*-V+(Y(
R)>Y(.))*V:IFFNK(R)THENY(R)=AY                PJ
·350 CHAR.,X(R),Y(R),"[RED]*":IFAX<>X(R)O
RAY<>Y(R)THENCHAR.,AX,AY," "                  AF
·360 IFFNCP(.)THENEXIT:ELSER=H-R              HO
·370 LOOPUNTILFNPD(.)>JD                      EN
```

```
·380 IFFNCP(.)THENFORA=.TOX:FORJ=VTOX:SOU
 NDT,P*X,V:COLOR5,J:CHAR.,X(.),Y(.),"*":N
 EXTJ,A:K=K-V:CHAR.,32+K*T,24," ":IFKTHEN
 GOSUB730:GOTO290:ELSEIFK=.THENRUN60     MH
·390 IFFNPD(.)=167THENPOKESM+X(.)+DX(A)+(
 DY(A)+Y(.))*UR,SP:SC=SC+L*50:SOUNDV,P*X*
 T,H,.,B,P*X,.:CHAR.,6,24,"[c 8]":PRINTUS
 INGF$;SC:GOTO320                        II
·400 IFFNPD(.)=155ORFNPD(.)=157THEN270   PK
·410 RN=FNPD(.)-176:A$="[s +]":POKE53265,
 11:FAST:GOSUB830                        JH
·420 ONRD(RN)GOSUB430,440,450,460:GOTO470 AN
·430 POKE2046,214:MOVSPR7,160,38:J=170:A=
 73:RETURN                               IB
·440 POKE2046,216:MOVSPR7,8,124:J=45:A=13
 2:RETURN                                LC
·450 POKE2046,216:MOVSPR7,312,124:J=295:A
 =132:RETURN                             DD
·460 POKE2046,214:MOVSPR7,160,214:J=170:A
 =200:RETURN                             GI
·470 SPRITE7,,V,.,V,V,.:SPRITEV,.,6,.,.,
 .,.:MOVSPRV,.#.:MOVSPRV,J,A:POKE2040,192
 :R=V:SPRITET,.,8:POKE2041,206:SPRITEX,.,
 V,.,.,.,.:MOVSPRX,173,140               HB
·480 FORJ=HTO6:MOVSPRJ,.#.:SPRITEJ,.,RN+V
 ,.,.,.,.,.:MOVSPRJ,IX(J),IY(J):NEXT     HK
·490 IFRN>6THENONRN-6GOTO520,530,550     JF
·500 IF(RNANDH)<HTHENA=((RNANDH)*T)+200:E
 LSEA=200                                GO
·510 POKE4600,4:POKE4601,4:U=X:W=45:Z=.:E
 =4:G=H:I=.005:S=(L<=H)*-T+(L>HANDL<=5)*-
 H+(L>5)*-4:FORJ=.TO4:POKE2041+J,A:NEXT:P
 OKE4595,A:GOTO560                       PG
·520 POKE4601,T:E=T:G=H:I=.:S=.:FORJ=HTO4
 :SPRITEJ,,12,,V,.:POKE2039+J,215:NEXT:FO
 RJ=5TO6:SPRITEJ,,12,,.,V:POKE2039+J,216:
 NEXT:POKE4595,215:POKE4600,80:GOTO560   HA
·530 POKE4601,T:E=T:G=H:U=T:W=180:Z=90:I=
 .:S=5:FORJ=HTO4:SPRITEJ,,10,V,V,.:POKE20
 39+J,214:NEXT:FORJ=5TO6:SPRITEJ,,10,V,.,
 V:POKE2039+J,216:NEXT:POKE4595,214      CJ
·540 GOTO560                             EF
·550 POKE4601,T:E=T:G=5:U=T:W=180:Z=.:I=.
 :S=5:FORJ=HTO4:SPRITEJ,,13,V,V,.:POKE203
 9+J,214:NEXT:FORJ=5TO6:SPRITEJ,,13,V,.,V
 :POKE2039+J,216:NEXT:POKE4595,214       AF
·560 FORA=.TO10:J=BUMP(V):NEXT:POKE53248+
 21,BM(RN):R=V:POKE4594,.:POKE4599,T     GE
·570 IF(RNANDH)<HTHENPOKE2047,210+(RNANDH
 ):ELSEPOKE2047,210                      KG
·580 SLOW:POKE53265,27                   AA
·590 REM MAIN LOOP                       DG
·600 DO:J=JOY(T):A=FNA(.):IFATHENR=A:POKE
 P,FNS(.)                                OK
·610 IFAANDA<>JTHENSPRITET,.:MOVSPRT,.#.:
 MOVSPRT,FNX(.),FNY(.):MOVSPRT,20;FND(.):
 MOVSPRT,FND(.)#X:POKEP+V,P(R):SPRITET,V:
 SOUNDV,P*X,SP,V,P*H,B*H,H               LN
·620 MOVSPRFNW(.),FNM(.)#S:S=S+I:J=BUMP(V
 ):LOOPUNTILFNC(J):ONFNC(J)GOTO660,630,60
 0                                       LH
·630 IFRN>H+HTHEN600:ELSEFORA=HTOX-T:IFFN
 B(J)THENSPRITET,.:SOUNDT,P*X,75,.,P*T,P,
 T:GOSUB700                              KB
·640 NEXT                                AE
·650 J=BUMP(V):J=BUMP(V):GOTO600         LC
·660 IFJ=B(7)+VTHENGOSUB710:GOTO650:ELSEI
 FJANDB(7)THEN650                        NL
·670 IFJ=B(6)+VTHENBM(RN)=PEEK(N):SCNCLR:
 POKEN,.:A$="[c +]":GOSUB750:GOTO300:ELSE
 IFJANDB(6)THEN650                       DB
·680 POKE4594,V:MOVSPRV,.#.:K=K-V:CHAR.,3
 2+K*T,24," ":SOUNDT,5000,150,T,,100,V:PO
 KEP,213:FORD=1TO3:FORJ=1TO16:SPRITEV,V,J
 :FORA=.TO20:NEXTA,J,D:BM(RN)=PEEK(N)    HK
·690 POKEN,.:IFKTHEN420:ELSERUN60        BK
·700 SC=SC+L*100:CHAR.,6,24,"[c 8]":PRINT
 USINGF$;SC:FORH=VTO5:FORT=VTO16:SPRITEA,
 V,T:NEXT:NEXT:T=2:H=3:SPRITEA,.:RETURN  EA
·710 SC=SC+L*1000:CHAR.,6,24,"[c 8]":PRIN
 TUSINGF$;SC:SOUNDT,P*T,50,V,B,B,T:POKEN,
 PEEK(N)ANDJD:SC$=SC$+CHR$(FNSC(.)):CHAR.
 ,16,24,"[YELLOW]"+SC$:RETURN            LA
·720 REM ERASE OLD CHARACTER POSITIONS   PA
·730 FORJ=.TOT:CHAR.,X(J),Y(J)," ":NEXT:R
 ETURN                                   BP
·740 REM DUNGEON CREATOR                 KC
·750 POKE53265,11:FAST:A$="[c +]":GOSUB83
 0:ONLGOSUB770,780,790,800,770,780,790,80
 0,790,810:CHAR.,T,5,S$:IFVAL(MID$(TR$,L,
 V))=LEN(SC$)THENCHAR.,H,5,"  "          LG
·760 POKE53265,27:SLOW:RETURN            GJ
·770 D=H:RN=4:FORJ=VTOH:D=D+V+(D=4)*4:RN=
 RN+V:CHAR.,J*10,J*4+T:GOSUB870:NEXTJ:RET
 URN                                     HE
·780 D=.:RN=.:FORJ=VTOH:FORA=VTOT:D=D+V+(
 D=4)*4:RN=RN+V:CHAR.,J*10-T,A*6+T:GOSUB8
 70:NEXTA,J:RETURN                       BM
·790 D=T:RN=7:FORJ=VTOT:FORA=VTOT:D=D+V+(
 D=4)*4:RN=RN+V+(RN=9)*9:CHAR.,J*12,A*6+T
 :GOSUB870:NEXTA,J:RETURN                CN
·800 D=V:RN=4:FORJ=VTO4:D=D+V+(D=4)*4:RN=
 RN+V:CHAR.,40-J*10,J*4+T:GOSUB870:NEXTJ:
 RETURN                                  NM
·810 D=.:RN=.:FORJ=.TO2:FORA=.TO2:D=D+V+(
 D=4)*4:RN=RN+V:CHAR.,J*10+7,A*7+4:GOSUB8
 70:NEXTA,J:RETURN                       NJ
·820 REM PRINT BORDER WITH A$            HG
·830 PRINTCHR$(147);C$(L);:FORJ=.TO79:PRI
 NTA$;:NEXT:FORJ=.TO19:PRINTA$A$TAB(38)A$
 A$;:NEXT:FORJ=.TO79:PRINTA$;:NEXT       JM
·840 PRINT"SCORE:[6"0"][3" "][[9" "]]  [c
 8]MEN [GREEN]";:FORJ=VTOK:PRINT"# ";:NE
 XT:CHAR.,12,24,"[WHITE]":PRINTL:CHAR.,16
 ,24,"[YELLOW]"+SC$:CHAR.,6,24,"[c 8]":PR
 INTUSINGF$;SC:RETURN                    HH
```

```
•850 REM PRINT ROOM W/DOOR AT CURRENT LOC
 ATION                                    DJ
•860 REM D=DOOR#,RN=ROOM#,C$(L)=COLOR     CF
•870 FORT=VTO4:A$(T)="[c +]":NEXT:T=2:A$(
 D)=CHR$(18)+CHR$(144)+CHR$(48+RN)+CHR$(1
 46)                                      BD
•880 PRINTC$(L);"[4"[c +]"]"A$(1);C$(L);"
 [4"[c +]"]";B$;"[c +]"T$;C$(L);"[c +]";B
 $;A$(2);T$;C$(L);A$(3);C$(L);B$;"[c +]"T
 $;C$(L);"[c +]";B$;"[4"[c +]"]"A$(4);C$(
 L);"[4"[c +]"]":RD(RN)=D:RETURN          MK
•890 REM VICTORY                          DL
•900 SOUND1,25000,300,1,300,3000,2:PRINT"
 [HOME][HOME][CLEAR]"TAB(4)"[8"[DOWN]"][Y
 ELLOW]YOU HAVE CONQUERED THE DUNGEON[3"!
 "]":SLEEP6:RETURN                        EB
•910 REM ARROW POINTERS                   BD
•920 DATA 206,207,208,209,206,207,208,209 JH
•930 REM LEVEL COLORS                     ID
•940 DATA "[BLUE]","[c 2]","[c 4]","[c 5]
 ","[CYAN]","[c 2]","[c 1]","[RED]","[c 7
 ]","[PURPLE]"                            HN
•950 REM JOYSTICK X&Y ADJUSTMENT DELTAS   GO
•960 DATA 0,-1,1,-1,1,,1,1,,1,-1,1,-1,,-1
 ,-1                                      MF
•970 REM INITIAL SPRITE POSITIONS         JI
•980 REM ERROR HANDLING                   EF
•990 SLOW:POKE53265,27:PRINTERR$(ER);" IN
  LINE ";EL:END                           ED
•1000 DATA 162,112,162,164,142,127,206,12
 7                                        IJ
•1010 REM REDEFINED CHARACTERS            KD
•1020 BANK15:SUM=0:FORI=2816TO2869:READJ:
 SUM=SUM+J:POKEI,J:NEXT:IFSUM<>7491THENPR
 INT"ERROR!":END:ELSESYS2816:POKE2604,PEE
 K(2604)AND240OR8                         DA
•1030 READA:IFA=-1THENRETURN:ELSEFORI=.TO
 7:READB:POKE8192+A*8+I,B:NEXT:GOTO1030   EL
•1040 DATA 169,,133,250,133,252,169,208   HJ
•1050 DATA 133,251,169,32,133,253,169,252 HA
•1060 DATA 141,185,2,169,16,141,54,11     PA
•1070 DATA 160,,169,250,162,14,32,116     AL
•1080 DATA 255,162,,32,119,255,200,208    OJ
•1090 DATA 241,206,54,11,240,7,230,251    DC
•1100 DATA 230,253,76,26,11,96            LJ
•1110 REM CHARACTER DATA                  JK
•1120 DATA35,60,126,219,255,255,219,102,6
 0                                        FL
•1130 DATA36,126,102,90,126,60,24,24,126  DK
•1140 DATA37,,56,16,56,68,68,56,.         KC
•1150 DATA38,,63,67,253,133,253,182,252   NF
•1160 DATA167,36,129,60,126,213,126,60,24 IP
•1170 DATA42,60,90,36,126,153,189,165,102 OJ
•1180 DATA91,,249,249,249,249,,191,191    GP
•1190 DATA102,63,,,243,243,,,063          KN
•1200 DATA155,255,128,191,128,159,128,143
 ,128                                     IN
•1210 DATA157,255,1,253,1,249,1,241,1     CP
•1220 DATA168,135,128,131,128,129,128,128
 ,255                                     FN
•1230 DATA169,225,1,193,1,129,1,1,255,-1  AE
•1240 REM SPRITE DATA                     FF
•1250 DATA0,8,,,28,,,62                   NF
•1260 DATA0,,8,,,8,,.                     IA
•1270 DATA127,,1,255,192,1,255,192        KP
•1280 DATA3,190,224,3,156,224,3,255       CF
•1290 DATA224,3,255,224,3,190,224,1       IH
•1300 DATA193,192,1,255,192,,127,.        GK
•1310 DATA0,,,,,,,.                       FG
•1320 DATA0,,,,,,,.                       EA
•1330 DATA0,,,,,,,.                       DC
•1340 DATA60,,,28,,,28,.                  DL
•1350 DATA127,36,1,255,192,1,255,192      FK
•1360 DATA3,190,224,3,156,224,3,255       BF
•1370 DATA224,3,255,224,3,190,224,1       PH
•1380 DATA193,192,1,255,192,,127,.        DK
•1390 DATA0,,,,,,,.                       KG
•1400 DATA0,,,,,,,.                       JA
•1410 DATA0,,,,,,,.                       IC
•1420 DATA0,,,,,,,.                       JM
•1430 DATA127,,1,255,192,1,255,192        NI
•1440 DATA3,190,228,3,156,230,3,255       KK
•1450 DATA255,3,255,230,3,190,228,1       EC
•1460 DATA193,192,1,255,192,,127,.        AK
•1470 DATA0,,,,,,,.                       PG
•1480 DATA0,,,,,,,.                       OA
•1490 DATA0,,,,,,,.                       NC
•1500 DATA0,,,,,,,.                       OM
•1510 DATA127,,1,255,192,1,255,192        CI
•1520 DATA3,190,224,3,156,224,3,255       LF
•1530 DATA224,3,255,224,3,190,224,1       JH
•1540 DATA193,192,1,255,192,,127,36       FE
•1550 DATA0,,28,,,28,,.                   MP
•1560 DATA60,,,,,,,.                      FG
•1570 DATA0,,,,,,,.                       EP
•1580 DATA0,,,,,,,.                       FJ
•1590 DATA127,,1,255,192,1,255,192        MJ
•1600 DATA3,190,224,3,156,224,3,255       IK
•1610 DATA224,3,255,224,3,190,224,1       NA
•1620 DATA193,192,1,255,192,,127,.        ND
•1630 DATA0,8,,,8,,,62                    BF
•1640 DATA0,,28,,,8,,.                    CH
•1650 DATA0,,,,,,,.                       JP
•1660 DATA0,,,,,,,.                       KJ
•1670 DATA127,,1,255,192,1,255,192        NJ
•1680 DATA3,190,224,3,156,224,3,255       PK
•1690 DATA224,3,255,224,3,190,224,1       IA
•1700 DATA193,192,1,255,192,18,127,.      NB
•1710 DATA28,,,28,,,30,.                  MH
•1720 DATA0,,,,,,,.                       MN
•1730 DATA0,,,,,,,.                       OP
•1740 DATA0,,,,,,,.                       PJ
•1750 DATA127,,1,255,192,1,255,192        GJ
•1760 DATA19,190,224,51,156,224,127,255   HJ
•1770 DATA224,51,255,224,19,190,224,1     MF
•1780 DATA193,192,1,255,192,,127,.        HD
```

```
•1790 DATA0,,,,,,,.                                    AD
•1800 DATA0,,,,,,,.                                    BK
•1810 DATA0,,,,,,30,.                                  IO
•1820 DATA0,28,,,28,,,18                               BC
•1830 DATA127,,1,255,192,1,255,192                     KC
•1840 DATA3,190,224,3,156,224,3,255                    GP
•1850 DATA224,3,255,224,3,190,224,1                    PJ
•1860 DATA193,192,1,255,192,,127,.                     GM
•1870 DATA0,,,,,,,.                                    HI
•1880 DATA0,,,,,,,.                                    GK
•1890 DATA0,,,,,,,.                                    GE
•1900 DATA0,,60,,,90,,.                                EH
•1910 DATA189,,,153,,,129,.                            ED
•1920 DATA0,129,,,66,,,36                              OE
•1930 DATA0,4,18,,2,9,,3                               MK
•1940 DATA4,128,7,2,64,14,1,32                         LP
•1950 DATA15,129,32,7,193,32,1,255                     CF
•1960 DATA32,,126,64,,63,128,.                         EI
•1970 DATA0,,,,,,,.                                    LE
•1980 DATA0,,30,,,45,,.                                LI
•1990 DATA94,128,,76,128,,64,128                       HM
•2000 DATA0,64,128,,33,,,18                            JE
•2010 DATA0,64,9,,128,4,128,112                        IL
•2020 DATA4,128,24,4,128,12,4,128                      FP
•2030 DATA28,4,128,15,12,128,7,255                     BO
•2040 DATA128,1,255,,,,,.                              EN
•2050 DATA0,,,,,,,.                                    CB
•2060 DATA0,16,,,80,,,80                               AF
•2070 DATA0,8,88,,26,116,,46                           CI
•2080 DATA82,60,74,89,255,154,95,255                   NC
•2090 DATA234,91,125,218,91,17,250,25                  HK
•2100 DATA255,154,8,127,18,8,99,18                     BN
•2110 DATA8,34,16,,20,16,,.                            OK
•2120 DATA0,,,,,,,.                                    IH
•2130 DATA0,,,,,,,.                                    HB
•2140 DATA0,,,8,,,10,16                                CB
•2150 DATA0,10,88,,26,116,,46                          PH
•2160 DATA82,60,74,89,255,154,87,255                   GH
•2170 DATA250,91,190,218,95,136,218,89                 JH
•2180 DATA255,152,72,126,16,72,195,16                  NE
•2190 DATA8,129,16,8,66,,,.                            KG
•2200 DATA0,,,,,,,.                                    NH
•2210 DATA3,255,,7,255,128,13,240                      DH
•2220 DATA64,,245,64,,240,70,.                         DJ
•2230 DATA240,73,,120,137,,125,153                     GI
•2240 DATA224,255,189,153,255,254,159,255              BL
•2250 DATA248,159,255,240,191,255,224,255              HK
•2260 DATA255,192,65,255,128,,255,192                  PE
•2270 DATA0,127,194,,127,195,,63                       LO
•2280 DATA227,,7,255,,3,254,.                          EG
•2290 DATA3,255,,7,255,128,13,240                      CH
•2300 DATA64,,245,64,,240,64,64                        IC
•2310 DATA240,64,160,120,142,144,125,153               DM
•2320 DATA152,255,185,159,255,249,191,255              NC
•2330 DATA253,127,255,254,63,255,248,31                AO
•2340 DATA255,192,1,255,128,,255,192                   LN
•2350 DATA0,127,192,,127,196,,63                       MK
•2360 DATA198,,15,254,,3,252,.                         LL
•2370 DATA0,,,,,,,.                                    GO
•2380 DATA0,,,,,,,.                                    HI
•2390 DATA0,,,8,,,8,.                                  MN
•2400 DATA0,8,,,8,,,8                                  PB
•2410 DATA0,,8,,,8,,.                                  AB
•2420 DATA8,,,8,,,,.                                   NJ
•2430 DATA0,,,,,,,.                                    IC
•2440 DATA0,,,,,,,.                                    JM
•2450 DATA0,,,,,,,.                                    LO
•2460 DATA0,,,,,,,.                                    MI
•2470 DATA0,,,,,,1,.                                   NE
•2480 DATA0,2,,,4,,,8                                  CD
•2490 DATA0,,16,,,32,,.                                IO
•2500 DATA64,,,,,,,.                                   AH
•2510 DATA0,,,,,,,.                                    NC
•2520 DATA0,,,,,,,.                                    OM
•2530 DATA0,,,,,,,.                                    AO
•2540 DATA0,,,,,,,.                                    BI
•2550 DATA0,,,,,,,.                                    AK
•2560 DATA0,,,,,,1,255                                 LE
•2570 DATA192,,,,,,,.                                  IC
•2580 DATA0,,,,,,,.                                    CN
•2590 DATA0,,,,,,,.                                    EP
•2600 DATA0,,,,,,,.                                    FJ
•2610 DATA0,,,,,,,.                                    EL
•2620 DATA0,,,,,,,.                                    EF
•2630 DATA0,,,,,,64,.                                  PO
•2640 DATA0,32,,,16,,,8                                HN
•2650 DATA0,,4,,,2,,.                                  MO
•2660 DATA1,,,,,,,.                                    HM
•2670 DATA0,,,,,,,.                                    JP
•2680 DATA0,,,,,,,.                                    KJ
•2690 DATA0,,,,,,,.                                    JL
•2700 DATA0,,,,,,,3                                    JA
•2710 DATA255,192,1,255,128,,231,.                     BH
•2720 DATA0,129,,,189,,,189                            CJ
•2730 DATA0,,126,,,60,,.                               NF
•2740 DATA24,,,24,,,24,.                               NC
•2750 DATA0,126,,1,255,128,,.                          LH
•2760 DATA0,,,,,,,.                                    PJ
•2770 DATA0,,,,,,,.                                    OL
•2780 DATA0,,,,,73,,.                                  HG
•2790 DATA0,,1,,64,,28,.                               BL
•2800 DATA1,62,64,,28,,,42                             FE
•2810 DATA0,,65,,,65,,.                                MH
•2820 DATA65,,,62,,,,.                                 KB
•2830 DATA0,,,,,,,.                                    BE
•2840 DATA0,,,,,,,.                                    FG
•2850 DATA0,,,,,,,.                                    EA
•2860 DATA0,,,,,,,15                                   GJ
•2870 DATA255,240,16,,72,36,146,136                    GG
•2880 DATA32,,136,36,66,152,32,160                     BI
•2890 DATA168,63,31,200,33,240,136,33                  MD
•2900 DATA176,136,33,240,144,32,,160                   IE
•2910 DATA63,255,192,,,,,.                             PL
•2920 DATA0,,,,,,,.                                    KG
•2930 DATA0,144,128,,73,,,144                          KE
•2940 DATA128,,73,,,,,.                                GJ
```

```
·2950 DATA127,,1,255,192,1,255,192      EG
·2960 DATA3,156,224,3,190,224,3,255     OH
·2970 DATA224,3,255,224,3,193,224,1     II
·2980 DATA190,192,1,255,192,,127,.      JJ
·2990 DATA0,,,,,,,.                     LE
·3000 DATA0,,,,,,,.                     PG
·3010 REM INTERRUPT DATA                AH
·3020 DATA120,169,25,141,20,3           NM
·3030 DATA169,12,141,21,3,173           FB
·3040 DATA248,17,133,251,169,1          BC
·3050 DATA133,252,32,157,13,88          IM
·3060 DATA96,173,242,17,240,3           NG
·3070 DATA76,252,12,198,251,208         DH
·3080 DATA27,173,248,17,133,251         MK
·3090 DATA165,252,24,109,243,17         GN
·3100 DATA174,249,17,157,249,7          KJ
·3110 DATA202,208,250,169,1,56          BE
·3120 DATA229,252,133,252,173,0         EO
·3130 DATA220,162,0,41,15,133           KP
·3140 DATA253,169,15,56,229,253         AL
·3150 DATA201,4,208,19,222,214          CN
·3160 DATA17,208,11,169,1,77            CL
·3170 DATA230,17,141,230,17,222         AH
·3180 DATA214,17,76,252,12,201          EE
·3190 DATA8,208,19,254,214,17           AE
·3200 DATA208,11,169,1,77,230           JE
·3210 DATA17,141,230,17,254,214         PG
·3220 DATA17,76,252,12,201,1            OH
·3230 DATA208,7,232,222,214,17          JM
·3240 DATA76,252,12,201,2,208           BG
·3250 DATA7,232,254,214,17,76           AB
·3260 DATA252,12,201,5,208,23           EP
·3270 DATA222,214,17,208,11,169         LI
·3280 DATA1,77,230,17,141,230           OF
·3290 DATA17,222,214,17,232,222         AP
·3300 DATA214,17,76,252,12,201          LM
·3310 DATA6,208,23,222,214,17           DI
·3320 DATA208,11,169,1,77,230           GM
·3330 DATA17,141,230,17,222,214         MA
·3340 DATA17,232,254,214,17,76          IA
·3350 DATA252,12,201,9,208,23           PI
·3360 DATA254,214,17,208,11,169         AE
·3370 DATA1,77,230,17,141,230           MK
·3380 DATA17,254,214,17,232,222         JF
·3390 DATA214,17,76,252,12,201          FH
·3400 DATA10,208,21,254,214,17          GE
·3410 DATA208,12,138,169,1,77           DJ
·3420 DATA230,17,141,230,17,254         NJ
·3430 DATA214,17,232,254,214,17         HE
·3440 DATA162,5,142,244,17,173          AE
·3450 DATA244,17,10,170,74,168          MH
·3460 DATA185,149,13,45,230,17          KA
·3470 DATA208,20,189,214,17,56          FA
·3480 DATA205,238,17,176,11,172         JE
·3490 DATA238,17,200,152,157,214        FG
·3500 DATA17,76,117,13,138,74           AC
·3510 DATA168,185,149,13,45,230         MN
·3520 DATA17,240,32,189,214,17          FE
·3530 DATA56,205,239,17,144,23          GE
·3540 DATA172,239,17,136,152,157        EA
·3550 DATA214,17,173,244,17,201         HD
·3560 DATA1,208,8,169,253,45            FB
·3570 DATA21,208,141,21,208,232         LJ
·3580 DATA189,214,17,56,205,240         CH
·3590 DATA17,176,11,172,240,17          GH
·3600 DATA200,152,157,214,17,76         GG
·3610 DATA117,13,189,214,17,56          IG
·3620 DATA205,241,17,144,23,172         DF
·3630 DATA241,17,136,152,157,214        KH
·3640 DATA17,173,244,17,201,1           MH
·3650 DATA208,8,169,253,45,21           IG
·3660 DATA208,141,21,208,206,244        PB
·3670 DATA17,48,3,76,1,13               JH
·3680 DATA32,196,13,238,46,208          PJ
·3690 DATA76,101,250,1,2,4              CO
·3700 DATA8,16,32,64,128,162            GI
·3710 DATA24,169,0,157,0,212            IO
·3720 DATA202,16,250,169,240,141        GE
·3730 DATA20,212,169,65,141,18          IO
·3740 DATA212,169,15,141,24,212         JP
·3750 DATA169,0,141,246,17,169          FD
·3760 DATA50,141,17,212,141,16          HM
·3770 DATA212,96,206,245,17,208         LI
·3780 DATA53,169,12,141,245,17          HH
·3790 DATA173,246,17,174,247,17         LJ
·3800 DATA56,221,11,14,144,5            PD
·3810 DATA169,0,141,246,17,174          GJ
·3820 DATA247,17,189,14,14,133          CL
·3830 DATA254,189,17,14,133,255         DA
·3840 DATA172,246,17,177,254,141        NH
·3850 DATA15,212,200,177,254,141        OC
·3860 DATA14,212,200,140,246,17         AD
·3870 DATA173,245,17,24,105,150         CG
·3880 DATA141,17,212,141,10,212         JH
·3890 DATA96,80,44,48,20,100            EN
·3900 DATA144,14,14,14,19,239           GF
·3910 DATA18,209,19,239,18,209          OO
·3920 DATA18,209,19,239,18,209          MI
·3930 DATA18,209,29,223,29,223          AM
·3940 DATA28,49,25,30,25,30             GF
·3950 DATA22,96,19,239,19,239           MI
·3960 DATA18,209,16,195,16,195          DG
·3970 DATA18,209,16,195,16,195          FI
·3980 DATA18,209,16,195,16,195          OK
·3990 DATA18,209,16,195,16,195          PM
·4000 DATA28,29,28,49,25,30             HC
·4010 DATA23,181,23,181,19,239          CK
·4020 DATA18,209,18,209,16,195          EK
·4030 DATA25,30,25,30,18,209            PF
·4040 DATA7,112,7,119,7,119             ND
·4050 DATA4,180,4,251,4,251             JP
·4060 DATA5,152,5,237,5,237             IB
·4070 DATA6,71,6,167,6,167              FO
·4080 DATA7,112,7,112,7,119             JI
·4090 DATA4,251,5,237,4,180             OA
·4100 DATA7,112,5,152,5,237             KG
```

```
•4110 DATA7,119,25,30,21,31                    OG
•4120 DATA18,209,25,30,21,31                   GN
•4130 DATA18,209,25,30,21,31                   NP
•4140 DATA18,209,25,30,21,31                   JB
•4150 DATA18,209,25,30,28,49                   HB
•4160 DATA29,223,28,49,22,96                   FI
•4170 DATA28,49,18,209,25,30                   GL
•4180 DATA21,31,18,209,25,30,25,30             CO
```

TIPS AHOY!

Continued from page 52

It's super easy to check the bytes free, and you can do it as often as you like. With the cursor resting on a blank line, press the back arrow key (←), then press RETURN. Bingo...there they are. —Buck Childress
Salem, OR

```
•10 PRINTCHR$(147)"LOADING DATA";:FORJ=70
 0TO753:READA:IFA<0ORA>255THEN30
•20 POKEJ,A:X=X+A:PRINT".";:NEXTJ:PRINT:P
 RINT:IFX=5958THEN40
•30 PRINT"ERROR IN DATA[3"."]":END
•40 PRINT"THE DATA IS LOADED,":PRINT
•50 PRINT"SYS 700 TO ACTIVATE AND DEACTIV
 ATE[3"."]":END
•60 DATA169,208,162,2,236,1,3,208,4,169,1
 39,162
•70 DATA227,141,0,3,142,1,3,96,173,0,2,20
 1
•80 DATA95,240,3,76,139,227,206,0,2,164,5
 2,165
•90 DATA51,56,229,49,176,1,136,170,152,56
 ,229,50
•100 DATA32,205,189,76,116,164
```

C-128 CENTER AND RIGHT JUSTIFY

You're probably feeling gypped out of a good command if you've tried to use the C-128's PRINT USING command to center strings on the screen. The documentation included with the *System Guide* is incorrect, depriving you of the use of a much needed command.

You'll have to define a control string, F$, with 40 ampersands to properly control this command, and suffix it with the appropriate control character, either "=" for centering or ">" for right justification. The control string lets the BASIC interpreter know what size of field you want your string to be centered in, and by changing the size of F$ you can also format text inside windows.

I am currently working on a quality word processor in BASIC 7.0, and encountered a great deal of frustration before I figured out through trial and error how this command worked. It was enormously useful in formatting text on the screen from the word processor. —Cleve Blakemore
Richmond, VA

```
•10 REM USE "=" TO CENTER STRING
•20 REM USE ">" TO RIGHT JUSTIFY
•30 PRINTCHR$(147)
•40 FORX=1TO39:F$=F$+"#":NEXT
•50 A$="CENTERED STRING"
•60 PRINTUSINGF$+"=";A$
•70 A$="RIGHT JUSTIFIED"
•80 PRINTUSINGF$+">";A$
•90 A$="LEFT JUSTIFIED"
•100 PRINT"[UP]";A$
```

BRANCH CALC

The BASIC language of the Commodore 64 is reasonably good. But there is still room for improvement. For example, if you've seen BASIC programs for other languages that offer calculated GOTOs and GOSUBs, chances are the program was much easier to understand. A GOSUB DRAWSCREEN is definitely preferable to GOSUB 13491. The 64 will possess this ability when you load and run *Branch Calc.*

Lines 120 and 130 transfer Kernal and BASIC ROM to RAM and protect it against RUN STOP/RESTORE. Since this takes some time, it you've already used a program that does this, RUN 140 to speed things up. —Shawn K. Smith
Bronx, NY

```
•100 REM* BRANCH CALC ---- SHAWN K. SMITH
•110 REM  "CALCULATED GOTOS AND GOSUBS"
•120 B=40960:K=57344:FORD=0TO8191:POKEB+D
 ,PEEK(B+D):POKEK+D,PEEK(K+D):NEXT
•130 POKE1,53:POKE64982,53
•140 S=48981:FORD=STOS+6:READY:POKED,Y
•150 NEXT:H=INT(S/256):POKE43170,H
•160 POKE43169,S-H*256:D=21:LOOP=180
•170 DATA 032,138,173,076,247,183,234
•180 D=D-1:PRINT"OK!":IFDTHEN  GOTO LOOP
```

...COMING IN THE SEPTEMBER ISSUE OF *AHOY!* (ON SALE JUNE 30)...

THE 128 V. THE CLONES
HOW DOES C-128 RATE AGAINST THE IBM-PC?

SUMMER CES REPORT
NEWS ON UPCOMING COMMODORE RELEASES

SPRITESHELL 128
FOR SPRIGHTLIER BASIC 7.0 GRAPHICS

DATA EAST BRINGS ARCADE REALISM HOME!



ENTER THE DATA EAST SWEEPSTAKES

OVER 100 PRIZES – Win free Data East Computer Games, a real Data East Arcade Game, and a trip for two to the National Arcade Expo!

No purchase necessary. See participating dealers for details. Void where prohibited.*

KARATE CHAMP® – Put on your black belt and challenge your friends or the computer through nine picturesque settings in this leading Martial Arts game to become the Karate Champ. For Commodore 64/128™ and Apple II® series.

*COMMANDO®** – As the crack shot Commando, battle overwhelming odds to defeat advancing rebel forces! Armed with only a machine gun and hand grenades, you must reach the fortress. For Commodore 64/128™, Apple II® series, and IBM.

*IKARI WARRIORS™** – Behind enemy lines, guerilla warfare are your tactics as you and your partner battle through the jungles, rivers, ruins, and opposing entrenchments in this interactive 2-player arcade hit. For Commodore 64/128™

BREAKTHRU™ – Your mission is to retrieve the secret fighter plane stolen by hostile renegades. Use your sophisticated assault vehicle to Break Thru each of the five enemy strongholds. For Commodore 64/128™

EXPRESS RAIDER™ – Bandits have seized the Gold City Express. Rescue the passengers, save the train, and recapture the loot! For Commodore 64/128™

TAG TEAM WRESTLING™ – Teamwork and stamina are the keys, as you and your partner battle your way through the Title Matches on your quest for the belt. For Commodore 64/128™, Apple II® series, and IBM.

*KUNG-FU MASTER®** – Battle the evil forces through the five dangerous floors in the wizard's castle to rescue the captive fair maiden. For Commodore 64/128™ and Apple II® series.

DATA EAST®

DATA EAST USA, INC. 470 NEEDLES DRIVE, SAN JOSE, CALIFORNIA 95112. (408) 286-7074.

Reader Service No. 141

ECHELON™



ACTUAL AMIGA GRAPHICS

3-D SPACE FLIGHT SIMULATOR

ECHELON . . . A top secret military facility where a few highly skilled pilots will be trained to operate the 21st Century's most awesome combat and exploration vehicle, the Lockheed C-104 Tomahawk. The C-104 spacecraft has the latest in advanced instrumentation, weaponry, and control systems.

ACTUAL C-64 SCREENS





Available Now For C-64/128.

Enhanced versions coming soon for APPLE, IBM, AMIGA, ATARI ST, and COMMODORE 128.



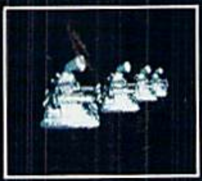


IT'S IMPRESSIVE ATTRIBUTES INCLUDE

- 4 Different "State of the Art" Weapons Systems
- Bi-Directional Transporter Unit
- Unmanned RPV (Remote Piloted Vehicle) Control
- Anti-Gravity Braking and Hovering
- Forward and Reverse Thrusters
- Twin 3-D Screens
- Real-Time Infrared Vector Graphics
- Non Real-Time Computer Enhanced Graphics
- Computer Assisted Control Systems
- Computer Information Data Link
- On Board Sub-Atomic Analyzer/Decoder

ACCESS™ Software Incorporated

ECHELON is a true 3-D space flight simulator that puts you at the controls of the C-104 Tomahawk and in the heart of the action. You'll experience the excitement and thrill of real spaceflight through superb three dimensional graphics. But, before you go exploring on your own, it would be wise to go through the carefully designed training simulators. These rigorous courses will develop your skills in manuevering the spacecraft, accurate selection and use of weapons, control of RPV's, object retrieval and refuel and docking procedures. Developing these skills are essential to your survival when encountering hostile enemy spacecraft or exploring unknown terrain. **ECHELON** is available now and offers you the Simulation Adventure of a Lifetime for only $39.95!!





If you can't find our products at your local dealer, you can order direct. For mail orders, enclose check or money order, plus $1.00 shipping and handling, and specify machine version desired. Order by phone on VISA, MasterCard or C.O.D. by calling TOLL FREE (800) 824-2549.
ACCESS SOFTWARE, INC. 2561 So. 1560 W. Wood Cross, UT 84087

Reader Service No. 161